

Remarks:

The Office Action of August 6, 2007 has been carefully reviewed, and this response addresses the Examiner's concerns.

I. STATUS OF THE CLAIMS

Claims 1-24 are pending in the application.

Claims 18-24 are rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Claims 1-2, 4-9 and 11-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Drisko et al. (US 5,933,523).

Claims 3 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Drisko et al. and Yokoi (US 2001/0022854).

II. OBJECTIONS TO THE SPECIFICATION

Applicants enclose herewith a copy of J. S. Lim, Two Dimensional Signal and Image Processing, ISBN 0-13-935322-4, pp. 476-94 as requested by the Examiner.

Applicants have also amended paragraph [0013] of the specification to remove the embedded hyperlinks.

Finally, Applicants have amended the Abstract to be fewer than 150 words in length.

III. THE 35 U.S.C. 101 REJECTION

Claims 18-24 are rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Applicants respectfully state that, for the reasons provided hereinbelow, claims 18-24 are patentable subject matter under 35 USC 101.

In In re Beauregard (53 F. 3d 1583 (Fed. Cir. 1995)), the Commissioner stated that computer program products embodied in a computer usable medium are patentable subject matter under 35 USC 101. *In re Beauregard* followed the decision in *In re Lowry* (32 F. 3d 1579

(Fed. Cir. 1994)) in which the Federal Circuit Court overturned a printed matter rejection for a claim for a memory including a data structure.

In *In re Lowry*, The Board of Patent Appeals and Interferences held that the memory containing stored information recited an article of manufacture but did not give patentable weight to the data structure. The Court found that the data structures were “specific electrical magnetic structure elements in a memory and provided tangible benefits;” as such the data structures were physical entities.

Following *In re Lowry* and *In re Beauregard*, a computer program product including a computer usable medium having computer readable code embodied therein is patentable subject matter and is covered by a statutory category.

The use of the Internet for distribution and execution of software is a relatively new technology. See for example, Microsoft’s decision to use the Internet for distribution of software as described in <http://www.microsoft.com/presspass/press/1996/may96/esdpr.msp> or the use of CORBA or DCOM for distributed software execution involving a transmission from server to client which can happen via a carrier wave (the CORBA and DCOM definitions date to the mid-1990s) or the use of applets (also dating to the mid-1990s).

The proposed interim guidelines stated that:

from a technological standpoint, a signal encoded with functional descriptive material is similar to a computer-readable memory encoded with functional descriptive material, in that they both create a functional interrelationship with a computer. In other words, a computer is able to execute the encoded functions, regardless of whether the format is a disk or a signal.

As such, a computer usable medium, having many possible embodiments, encoded with computer-readable code is a specific electrical/magnetic structure and provides a tangible result and, according to *In re Lowry* and *In re Beauregard*, is patentable subject matter and falls under a statutory category.

Assuming, arguendo, that a computer usable medium as stated in *In re Beauregard* is only statutory for devices such as floppy disks, rigid magnetic disks, magnetic tape, optical tape, optical recording discs, punched tape or cards, or the like results in legal contradictions as presented herein below.

A. Taking arguendo the interpretation proposed by the Examiner, a computer usable memory has non-statutory equivalents under the Doctrine of Equivalents. (If a claimed invention is statutory, can the equivalents - that provide the same function, in the same way, to obtain the same result - be nonstatutory?)

As the proposed guidelines state, a carrier wave having computer readable code embodied therein serves the same purpose, creating a functional interrelationship with a computer, as a computer readable memory and obtains the same result, the computer is able to execute the encoded functions. Also, a carrier wave having computer readable code embodied therein performs that same purpose in the same way as a computer readable memory having computer readable code embodied therein; both carry bits of information and both operate as communication channels (see for example, Wolf, J.K., *Magnetic recording as a communications channel*, 1994 IEEE International Symposium on Information Theory, 1994. Proceeding, 27 June-1 July 1994 Page(s):5 or Moon, J, *Signal-to-noise ratio definition for magnetic recording channels with transition noise*, IEEE Transactions on Magnetics, Volume 36, Issue 5, Part 2, Sept 2000 Page(s):3881, 3883).¹ The carrier wave having computer readable code embodied therein obtains the same result, i.e. to enable a computer to provide a useful result by executing the code, as a computer readable memory having computer readable code embodied therein. An accused product or process may infringe a patent either literally or under the Doctrine of Equivalents. To literally infringe a patent claim, the product or process at issue must include each and every element of the claim. *Builders Concrete, Inc v. Bremerton Concrete Products*, 757 F.2d 255, 257 (Fed. Cir. 1985). In the event of failing to meet the standards established for demonstrating a *prima-facie* case of literal infringement, the analysis then proceeds to another theory of infringement – the Doctrine of Equivalents.

The Doctrine of Equivalents enables protect patent owners relief against infringers' making "unimportant and insubstantial changes" to a patented invention which "though adding nothing, would be enough ... [to evade] the reach of the law." *Graver Tank & Mfg. Co. v. Linde Air Products Co.*, 339 U.S. 605, 606, 70 S.Ct. 854, 856, 94 L.Ed. 1097 (1950). The doctrine may

¹ Copies of the above cited pages of the Wolf and Moon articles are provided herewith in the Appendix and pertinent passages are highlighted.

result in a virtual expansion of the scope of a patentee's claims in certain circumstances where, although it is shown that the literal language of a claim's elements is not met by an accused product, it may be proven that the accused product, analyzed on an element by element basis, "performs substantially the same overall function or work, in substantially the same way, to obtain substantially the same overall result as the claimed invention." *Pennwalt Co. v. Durand-Wayland Inc.*, 833 F.2d 931, 934 (Fed. Cir. 1987).

Therefore, infringement under the Doctrine of Equivalents requires that each corresponding element of the accused apparatus, device, composition or process, that is not literally met by a limitation of a claim, performs substantially the same function in substantially the same way to yield the same, or substantially the same, result as each corresponding limitation of the claim. *Graver Tank & Mfg. Co., Inc. v. Linde Air Products, Co.*, 339 U.S. 605; *Perkin-Elmer Co. v. Computervision Corp.* 732 F.2d 888, (Fed. Cir. 1984), *cert. denied*, 469 U.S. 857 (1984); *Pennwalt Co. v. Durand-Wayland, Inc.*, 833 F. 2d at 934 (Fed. Cir. 1987), *cert. denied*, 485 U.S. 961 (1988), and *cert. denied*, 485 U.S. 1009 (1988).

The use of Doctrine of Equivalents can also be further burdened by file wrapper estoppel (*Warner-Jenkinson Co. v. Hilton-Davis Chem. Co.*, 520 U.S. 17 (1997) and *Festo Corp. v. Shoketsa Kinzoky Kogyo Kaboshiki Co. Ltd.*, 535 U.S. ____ (2002); alleged disclaimers (*Johnson & Johnson Assocs. v. U.R.E. Serv. Co.*, 285 F.3d 1046 (Fed. Cir. 2002) (*en banc*); and by prior art limits (*Wilson Sporting Goods Co. v. David Geoffrey Assoc.*, 904 F.2d 677 (Fed. Cir. 1990) and if and when "means for" limitations per 35 U.S.C. § 112 (6th par.) are deemed to be involved, by the interaction of that paragraph with the Doctrine of Equivalents as explained, e.g., in *Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308 (Fed. Cir. 1999).²

As stated above and considering the law summarized here, a claim allowed before the mid-1990s reciting a computer usable memory having computer readable code embodied therein might protect against an infringing product utilizing a carrier wave (such as the Internet) to perform the same function since such a product would infringe under the Doctrine of Equivalents

² In *Al-Site* important distinctions are made between before-arising and after-arising technologies that can affect initial claim construction and can affect permissible range of application of the Doctrine of Equivalents. That distinction may also be significant in Doctrine of Equivalents consideration, even if no "means for..." limitation is involved.

unless subject to one or more of the other Doctrine of Equivalents limitations. Since the carrier wave is an after developed technology, limitations of the Doctrine of Equivalents place by *Warner-Jenkinson* and *Festo* and preceding cases might not apply to pre mid-1990s while those limitations of the Doctrine of Equivalents would apply to the Applicant's claims.

Therefore, the application of the proposed guidelines to claims for computer program product comprising computer readable medium would render effective coverage of claims dependent on when the claim was written. This is an anomaly inconsistent with the clarifying purpose of the proposed guideline. Furthermore, the anti-carrier wave feature of the proposed guidelines presents a situation where, if the applicant does not explicitly list that a carrier wave is a computer usable medium, a claim for a computer usable medium, which is identical to the claims in *In re Beauregard* would protect against products using a carrier wave to embody computer readable code by means of the Doctrine of Equivalents while applicants who are more explicit in their definition of a computer usable medium would have their claims rejected, rendered patentable only by a narrowing amendment and therefore would be blocked by File Wrapper Estoppel from access to the Doctrine of Equivalents under *Festo*.

The application of the Annex IV of the proposed guidelines to claims for computer program products comprising a computer usable medium presents a novel situation in patent law where a bona fide equivalent (under the Doctrine of Equivalents) is declared to be nonstatutory. Such a situation would be a case of first impression and is counter to applying the known law for determining compliance with 35 USC 101.

B. The Cross Border Infringement Issue

In *NTP Inc. v. Research in Motion, Inc.*, 418 F.3d 1282, 75 USPQ 2d 1763 (Fed. Cir. 2005) it was well explained that the steps of utilization of the Blackberry® system occurring partially in Canada did not infringe method claims of the patents in suit but that such usage did infringe apparatus (system) claims. 75 USPQ 2d at 1786-93. Nor did the defendant (nor customers of defendant induced by defendant) infringe by "exports" of text and voice message encoded signals to Canada under 35 U.S.C. § 271(f) [75 USPQ 2d at 1793-94] nor commit import of product-by-process infringement under 35 U.S.C. § 271(g) [75 USPQ 2d at 1794-95].

Since under *NTP v RIM* methods claims do not provide protection against extra-territorial infringers, Applicant's protection would only be based on the system claim that is practiced by the user not the extra-territorial provider of the software.

On the other hand in *Eolas Tech Inc. v. Microsoft Corp.*, 399 F.3d 1375, 73 USPQ 2d 1782 (Fed. Cir. 2005) and *AT&T Corp. v. Microsoft Corp.*, 414 F.3d 1366, 75 USPQ 2d 1506 (Fed. Cir. 2001) shipment of software on disks and indeed on a master disk and also as signals to Europe to be copied, with copies to be bundled with European made, sold, used computers did infringe a U.S. patent under the 35 U.S.C. § 271(f) export provisions. In *AT&T*, the Federal Circuit observed (75 USPQ 2d at 1509-10):

“Were we to hold that Microsoft’s supply by exportation of the master versions...avoids infringement we would be subverting the remedial nature of § 271(f), permitting a technical avoidance of the statute by ignoring the advances in a field of technology – and its associated industry practices – that developed after the enactment of § 271(f). It would be unsound to construe a statutory provision that was originally enacted to encourage advances in technology by closing a loophole, in a manner that allows the very advances in technology thus encouraged to subvert that intent. Section 271(f), if it is to remain effective must therefore be interpreted in a manner that is appropriate to the nature of the technology at issue....”

The Federal Circuit’s decision is consistent with a situation of streaming code abroad via the Internet rather than shipping a master disk. As stated by the Federal Circuit, “Additionally, we cannot accept Microsoft’s suggestion that software sent by electronic transmission must be treated differently for purposes of § 271(f) liability from software shipped on disks, see Tr. of Dec. 12, 2003 Hearing, at 8:8-17 (J.A. 351), as it would amount to an exaltation of form over substance. “*ATT v. Microsoft*, 414 F. 3d 1366 (2005).

A petition for *Certiori* was granted October 28, 2006 and Federal Circuit decision was reversed on other grounds. *Microsoft v. ATT*, 550 U. S. ____ (2007), Decided April 30, 2007. As stated in the U.S. Supreme Court decision, “Until it is expressed as a computer-readable .copy,. e.g., on a CD-ROM, Windows software indeed any software detached from an activating medium remains uncombinable.” *Id.*

The patent in question in *ATT v. Microsoft*, RE32580, was re-issued in 1988 before the *In re Lowry* and in *In re Beauregard* decisions and therefore does not have a *Beauregard* type

claim. A Beauregard type claim would have avoided the controversy since, in claiming the software as being embodied in a computer readable medium (copy), the master disk (or the signal transmission, using the Federal Circuit statement) would have been the claimed invention and Microsoft would have been shipping an infringing product.

Under the proposed Interim guidelines, transmission of software overseas to be bundled with a European made computer would not infringe while shipping a physical medium with the claimed software embodied therein would infringe. Therefore, the proposed Interim guidelines run counter to the statement by the Federal Circuit and permit "a technical avoidance of the statute by ignoring the advances in a field of technology." Also, in the words of the Federal Circuit, the proposed guidelines place to form over substance allowing infringers to use this form over substance guidelines to infringe without consequence.

Should the Examiner apply guidelines not present in the MPEP to enable infringers to escape the statute by using advances in a field of technology that produce equivalents to the claimed invention?

Applicants respectfully state that such a rejection would be counter to the Constitutional basis of the patent statute.

For the reasons presented hereinabove, Applicants respectfully state that a utility rejection of claims 18-24 is not proper and should be withdrawn.

IV. THE 35 U.S.C. 102(b) REJECTION

Claims 1-2, 4-9 and 11-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Drisko et al. (US 5,933,523)(the '523 patent).

Applicants respectfully assert that the claimed inventions are not anticipated by the '523 patent since the '523 patent does not disclose culling the detected edges, detecting region corner points based on a predetermined relationship between each candidate corner point and characteristic edge points, and does not disclose obtaining a measure of cornerness (cornerness, also referred to as corner strength, is a term of art having a well known meaning for those skilled in the art) as detailed in the remarks given below. Regarding claim 11, Applicants respectfully state that the limitations of claim 11 invoke 35 USC 112 paragraph 6; Applicants respectfully

state that the '523 patent does not disclose structures equivalent to the structures in the specification corresponding to the functions in the "means plus function" limitations of claim 11.

Regarding claims 1 and 18, the Examiner identifies the third limitation, "culling the detected edges in order to obtain a reduced edge group from the detected edges, the reduced edge group comprising a plurality of points," with Figures 4C and 4D and block 503 in Figure 5- Remove Known-Nozzle Points of the '523 patent.

In the '523 patent, the operation at block 503 in Figure 5 is described in col. 9, lines 23-32 as follows:

At 503, nozzle information is used to remove corner points corresponding to known nozzle points. Nozzle information may be in the form of a model, or a list of points in the physical or image space. Conventional techniques such as masking or point-by-point comparison may be used to identify and remove corner points corresponding to the nozzle at this stage. Similarly, information related to other objects known to show up in the image may be used to remove corner points believed to correspond to those objects.

Applicants respectfully state that the operation block 503 in figure 5 up to '523 patent discloses removing corner points but does not disclose "culling the detected edges", which is a limitation of claim 1.

The Examiner identifies "detecting region corner points ...based on a predetermined relationship," another limitation of claim 1, with block 502 of Figure 5 of the '523 patent. In the '523 patent, the extent coordinate system is described, in col. 7, lines 2739, as a linear transformation indicative of rotation, as shown below:

At 501, the list of boundary points ($x_{sub.i}$, $y_{sub.i}$) are transformed from the image coordinate space into an extents coordinate space ($x_{sub.Ei}$, $y_{sub.Ei}$). The extents coordinate space is shown in FIG. 6C as a coordinate space which is fixed in relation to the coordinate space corresponding to the initial expected device orientation. Therefore, the axes ($X_{sub.E}$, $Y_{sub.E}$) of the extents coordinate space are offset by a fixed angle, for example, 45.degree., from the axes ($X_{sub.p}$, $Y_{sub.p}$) of the coordinate space corresponding to the image-space presentation angle (determined by mapping the physical space presentation angle via the calibration) representing the initial expected device orientation as it appears in the image. Because the relationship between the image coordinate space and the initial expected device orientation is described by the presentation angle and the relationship between the initial expected device orientation and the extents

coordinate space is fixed, a relationship between the image coordinate system and the extents coordinate system can be determined and described using common trigonometric formulas.

In a linear transformation (in fact, in any one-to-one transformation whether linear or not) points in one space are mapped into correspondent points in the transformed space. Therefore, corner points in the image coordinate space are mapped into corresponding corner points in the extent coordinate space.

As disclose in the '523 patent, in col. 8, lines 46-67 and in col. 9, lines 1-22 and in Figure 6D:

At 502, the boundary points transformed at 501 are evaluated to determine maximum (x.sub.max, y.sub.max) and minimum (x.sub.min, y.sub.min) values with respect to each of extents axes X.sub.E and Y.sub.E, which values define the extents. These extents are determined by minimum and maximum x and y positions of all the boundary points in the extents coordinate space. Both the extents and the extents points may be determined by comparing and updating the maximum and minimum values (x.sub.max, y.sub.max, x.sub.min, and y.sub.min) with each boundary point, x.sub.Ei, y.sub.Ei. The resulting extents points are shown, for example, as E1-E4 (in FIG. 6D). Extents points in the extents coordinate space correspond to corner points in the image coordinate space.

Indices may be associated and updated with the extents points. Indices may be used to determine corner points within the image coordinate space corresponding to the extents points without requiring any further computation. For example, the following algorithm may be used to compute extents points and update corresponding indices:

```
if  $x_{min} > x_{Ei}$ 
   $x_{min} = x_{Ei}$ 
  minX_i = i
else if  $x_{max} < x_{Ei}$ 
   $x_{max} = x_{Ei}$ 
  maxX_i = i
if  $y_{min} < y_{Ei}$ 
   $y_{min} = y_{Ei}$ 
  minY_i = i
else if  $y_{max} < y_{Ei}$ 
   $y_{max} = y_{Ei}$ 
  maxY_i = i.
```

In this algorithm, x.sub.min, x.sub.max, y.sub.min and y.sub.max correspond to extents in the extents coordinate space. The index maxX.sub.-- i corresponds to the upper right corner of the chip, that given by minX.sub.-- i to the lower left corner, that given by maxY.sub.-- i to the upper left, and the that given by

minY.sub.-- i to the lower right. Using these relationships, corner points in the image coordinate space may be determined based on the extents points in the extents coordinate space without performing complex complete mapping or rotating operations. Alternatively, if the extents points are calculated using relationships (1)-(8) without using indexes, the extents points of the extents coordinate space may be translated into corner points of the image coordinate space using trigonometric relationships similar to relationships (1)-(8).

In the above algorithm from the '523 patent, "extents" define the extent of the boundary of the device and corner points are mapped to corner points in the extent coordinate space and given indices so that corner points can be identified by the index without requiring the transformation.

Applicants respectfully state that the 523 patent does not disclose "detecting region corner points from the plurality of candidate corner points based on a predetermined relationship between each of the candidate corner points and characteristic edge points," a limitation of claim 1.

Regarding claims 4 and 14, Applicants defined in the specification, in paragraph 13, cornerness as being the same as the term "corner strength" ("the measure of *cornerness* is also referred to as a measure of corner strength"). "Corner strength" is a term of art which has a well-defined meaning in the image processing/Computer vision field. (A number of references are provided in the Appendix showing widespread use of the term "corner strength" and in, some instances, the equivalent use of the term "cornerness.") Applicants respectfully state that the '523 patent does not apply or disclose the use of the concept of corner strength as would be interpreted by one of ordinary skill in the art of image processing/computer vision.

The first step in determining whether a claim is anticipated, or is obvious in view of prior art, is to interpret the claim. ("It is elementary in patent law that, in determining whether a patent is valid, the first step is to determine the meaning and scope of each claim in suit."

Lemelson v. Gen. Mills, Inc., 968 F.2d 1202, 1206, 23 U.S.P.Q.2D (BNA) 1284, 1287 (Fed. Cir. 1992).) When not defined by applicant in the specification, the words of a claim must be read as they would be interpreted by those of ordinary skill in the art. (MPEP 211.01) (*Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1342, 60 USPQ2d 1851, 1854 (Fed. Cir. 2001) ("explaining the court's analytical process for determining the meaning of disputed claim terms")). Applicants

respectfully state that the term “cornerness” should be read as used in this specification and as would be understood by one skilled in the art, as evidenced by the references provided in the Appendix.

The Examiner identified col. 6, lines 42-50 of the ‘523 patent as disclosing “obtaining a measure of cornerness,” a limitation of claims 4 and 14. Col. 6, lines 42-50 of the ‘523 patent disclose:

As described in more detail with respect to FIGS. 5 and 6A-6F, the illustrated embodiment specifically determines correspondence between the boundary features output at 302 and the corners of the generally rectangular device. For instance, as shown in FIG. 4D, the boundary features in FIG. 4C (output of the processing at 302) corresponding to the corners of the generally rectangular device are identified. As shown, only three of the boundary features (E1, E2, E3) from the processing at 302 are identified as corresponding to the corners of the generally rectangular device since the boundary features most closely corresponding to the fourth corner actually correspond to the nozzle (depicted as 402A in FIG. 4A).

Applicants respectfully state that col. 6, lines 42-50 of the ‘523 patent do not disclose “obtaining a measure of cornerness” as would be understood by one of ordinary skill in the art.

The Examiner identifies col. 6, line 5 of the ‘523 patent (“Once corresponding features are identified, they are fit against model input 414D to determine generally rectangular device position at 303B (same as 506 in FIG. 5), as shown in FIG. 4E.”) as disclosing “selecting the plurality of candidate corner points from the plurality of point by applying a predetermined criterion,” a limitation of claims 4 and 14. Applicants respectfully state that col. 6, line 5 of the ‘523 patent disclose determining generally rectangular device position which does not correspond to a claim limitation of claims 4 and 14.

Regarding claims 5, 13 and 20, the comments provided above regarding the second limitation of claims 1 and 18 can be applied. Based on those comments, Applicants respectfully state that the 523 patent does not disclose the added limitations of claims 5, 13 and 20.

Regarding claims 8, 16 and 23, based on the comments made above, Applicants respectfully state that the 523 patent does not disclose the added limitation of claims 8, 16 and 23.

Regarding claim 11, the limitations of claim 11,

means for detecting edges interior to a region of interest;

means for culling the detected edges in order to obtain a reduced edge group from the detected edges, the reduced edge group comprising a plurality of points;

means for selecting a plurality of candidate corner points from the plurality of points; and,

means for identifying region corner points from the plurality of candidate corner points based on a predetermined relationship between each candidate corner point and characteristic edge points of the region of interest;

invoke 35 USC 112, paragraph 6.

The MPEP in section 2182 details how to interpret (construe) a limitation that invokes 35 U.S.C. 112, sixth paragraph. In order to construe a limitation invoking 35 U.S.C. 112, sixth paragraph, first, the function in the means plus function limitation has to be construed, and, second, the structures in the specification corresponding to the construed function have to be identified (MPEP 2182).

Paragraph 13 and paragraph 15 of the Applicants' specification detail the structures corresponding to the recited functions in claim 11. Applicants respectfully state, by the reasons given above, that the '523 patent does not disclose structure is equivalent to the corresponding structures in the Applicants' specification.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); MPEP 2131.

Claims 2, 4-9 are dependent on claim 1, claims 12-17 are dependent on claim 11 and claims 19-24 are dependent on claim 18.

Applicants respectfully state that claims 1-2, 4-9 and 11-24 are not anticipated by the '523 patent.

V. THE 35 U.S.C. 103(a) REJECTION

Claims 3 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Drisko et al. and Yokoi (US 2001/0022854)(the '854 publication).

As stated above, the '523 patent (Drisko et al.) does not disclose all the limitations of claim 1. Claims 3 and 10 are dependent on claim 1. Applicants respectfully state that the Yokoi (the '854 publication) does not disclose the limitations of claim 1 that are not disclosed by the '523 patent.

Since the '523 patent and Yokoi, either separately or in combination, do not teach or suggest all the limitations of claims 3 or 10, applicants respectfully state that a prima facie case of obviousness has not been established.

VI. CONCLUSION

In view of the above amendments and remarks, Applicants assert that claims 1-24 are now in condition for allowance and respectfully request the Examiner to pass the case to issue.

The Director of Patents and Trademarks is authorized to charge any fee deficiencies, or to credit any overpayments, to Deposit Account No. 03-2410, Order No. 12078-205.

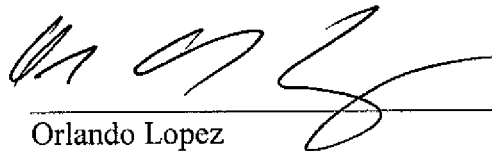
In accordance with Section 714.01 of the M.P.E.P., the following information is presented in the event that a call may be deemed desirable by the Examiner:

ORLANDO LOPEZ (617) 345-3000

Respectfully submitted,
Lawrence E. Albertelli et al., Applicants

Dated: November 2, 2007

By:



Orlando Lopez
Reg. No. 46,880
Attorney for Applicants

TWO-DIMENSIONAL SIGNAL AND IMAGE PROCESSING

JAE S. LIM

Department of Electrical Engineering
and Computer Science
Massachusetts Institute of Technology



P T R PRENTICE HALL, Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Lim, Jae S.

Two-dimensional signal and image processing / Jae S. Lim.
p. cm. — (Prentice Hall signal processing series)

Bibliography: p.

Includes index.

ISBN 0-13-935322-4

1. Signal processing—Digital techniques. 2. Image processing—
Digital techniques. I. Title. II. Series.

TK5102.5.L54 1990

621.382'2—dc20

89-33088

CIP

Editorial/production supervision: *Raeia Maes*

Cover design: *Ben Santora*

Manufacturing buyer: *Mary Ann Gloriande*



© 1990 by P T R Prentice-Hall, Inc.

A Simon & Schuster Company

Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be
reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4

ISBN 0-13-935322-4

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

by wideband Gaussian random noise at an SNR of 7 dB. The degraded image is shown in Figure 8.21(b). Figure 8.21(c) shows the image processed by a separable median filter with a window size of 3 for both the horizontal and vertical 1-D median filters. Although the very sharp edges are not blurred, median filtering blurs the image significantly. In the second example, the original image from Figure 8.21(a) is degraded by salt-and-pepper noise. The degraded image is shown in Figure 8.22(a) and the image processed by the same separable median filter used in Figure 8.21 is shown in Figure 8.22(b). This example shows that median filtering is quite effective in removing salt-and-pepper noise.

8.2.3 Out-Range Pixel Smoothing

Like median filtering, *out-range pixel smoothing* is a nonlinear operation and is useful in reducing salt-and-pepper noise. In this method, a window slides along the image, and the average of the pixel values, excluding the pixel being processed, is obtained. If the difference between the average and the value of the pixel processed is above some threshold, then the current pixel value is replaced by the average. Otherwise, the value is not affected. Because it is difficult to determine the best parameter values in advance, it may be useful to process an image using several different threshold values and window sizes and select the best result.

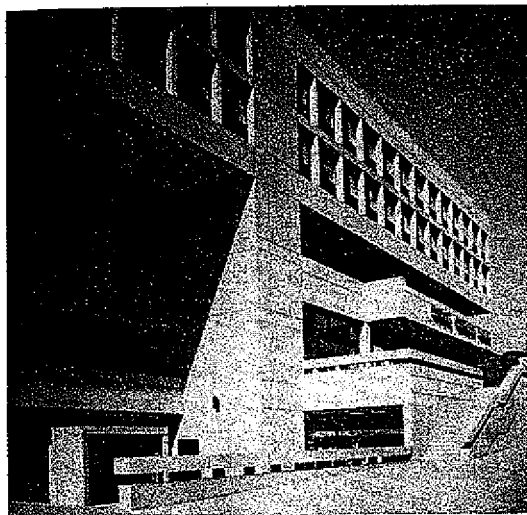
Figure 8.23 illustrates the performance of out-range pixel smoothing. The image in Figure 8.23 is the result after processing the image in Figure 8.22(a) using out-range pixel smoothing with a threshold value of 50 and a 3×3 -point window.

8.3 EDGE DETECTION

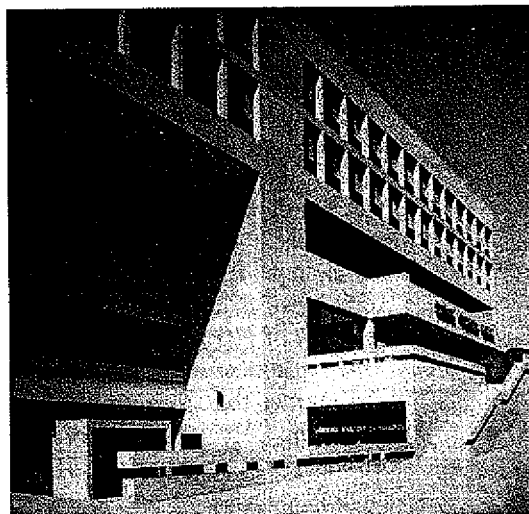
An *edge* in an image is a boundary or contour at which a significant change occurs in some physical aspect of an image, such as the surface reflectance, illumination, or the distances of the visible surfaces from the viewer. Changes in physical aspects manifest themselves in a variety of ways, including changes in intensity, color, and texture. In our discussion, we are concerned only with the changes in image intensity.

Detecting edges is very useful in a number of contexts. For example, in a typical image understanding task such as object identification, an essential step is to segment an image into different regions corresponding to different objects in the scene. Edge detection is often the first step in image segmentation. As another example, one approach to the development of a low bit-rate image coding system is to code only the detected edges. It is well known that an image that consists of only edges is highly intelligible.

The significance of a physical change in an image depends on the application; an intensity change that would be classified as an edge in some applications might not be considered an edge in other applications. In an object identification system, an object's boundaries may be sufficient for identification, and contours that rep-



(a)



(b)

Figure 8.22 Example of salt-and-pepper noise reduction by median filtering. (a) Image in Figure 8.21(a) degraded by salt-and-pepper noise; (b) processed image by the same separable median filter used in Figure 8.21.

resent additional details within the object may not be considered edges. An edge cannot be defined, then, outside of the context of an application. Nevertheless, edge detection algorithms that detect edges that are useful in a broad set of applications have been developed. In this section, we discuss some of the more representative edge detection algorithms.

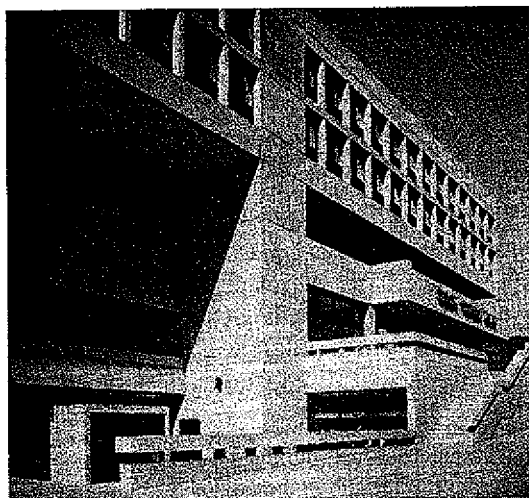


Figure 8.23 Example of salt-and-pepper noise reduction by outrange pixel smoothing. Image in Figure 8.22(a) processed by outrange pixel smoothing with threshold value of 50 and window size of 3×3 .

8.3.1 Gradient-Based Methods

Consider an analog* function $f(x)$ which represents a typical 1-D edge, as shown in Figure 8.24(a). In typical problems, it is reasonable to consider the value x_0 in the figure an edge point. One way to determine x_0 is to compute the first derivative $f'(x)$ or the second derivative $f''(x)$. Figures 8.24(b) and (c) show $f'(x)$ and $f''(x)$. From the figure, the value x_0 can be determined by looking for the local extremum (maximum or minimum) of $f'(x)$ or by looking for a zero crossing of $f''(x)$ where $f''(x)$ changes its sign. In this section, we discuss methods that exploit the characteristics of $f'(x)$. In the next section, we discuss methods that exploit the characteristics of $f''(x)$.

In addition to determining the possible edge point x_0 , $f'(x)$ can also be used in estimating the strength and direction of the edge. If $|f'(x)|$ is very large, $f(x)$ is changing very rapidly and a rapid change in intensity is indicated. If $f'(x)$ is positive, $f(x)$ is increasing. Based on the above observations, one approach to detecting edges is to use the system shown in Figure 8.25. In the system, first $|f'(x)|$ is computed from $f(x)$. If $|f'(x)|$ is greater than some threshold, it is a candidate to be an edge. If all values of x such that $|f'(x)|$ is greater than a certain threshold are detected to be edges, an edge will appear as a line rather than a point. To avoid this problem, we further require $|f'(x)|$ to have a local maximum at the edge points. It may be desirable to determine whether $f(x)$ is increasing or decreasing at $x = x_0$. The necessary information is contained in $f'(x)$ at $x = x_0$. The choice of the threshold depends on the application. As the threshold increases, only the values of x where $f(x)$ changes rapidly will be registered as candidate edges. Since it is difficult to choose the threshold optimally, some trial and error is usually involved. It is also possible to choose the threshold adaptively. The system in Figure 8.25 is based on the particular type of edge shown in Figure 8.24(a), but it is generally applicable to detecting various other types of edges.

The generalization of $f'(x)$ to a 2-D function $f(x, y)$ is the gradient $\nabla f(x, y)$ given by

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y \quad (8.6)$$

where \hat{i}_x is the unit vector in the x-direction and \hat{i}_y is the unit vector in the y-direction. A generalization of the edge detection system in Figure 8.25 based on $\nabla f(x, y)$ is shown in Figure 8.26. The magnitude of $\nabla f(x, y)$ is first computed and is then compared with a threshold to determine candidate edge points. If all values of (x, y) such that $|\nabla f(x, y)|$ is greater than a certain threshold are detected to be edges, the edges will appear as strips rather than lines. The process of determining an edge line from a strip of candidate edge points is called *edge thinning*. In one simple edge thinning algorithm, the edge points are selected by

*Sometimes, it is more convenient to develop results in the analog domain. In such instances, we will begin the development of results in the analog domain and then discretize the results at some later point in the development.

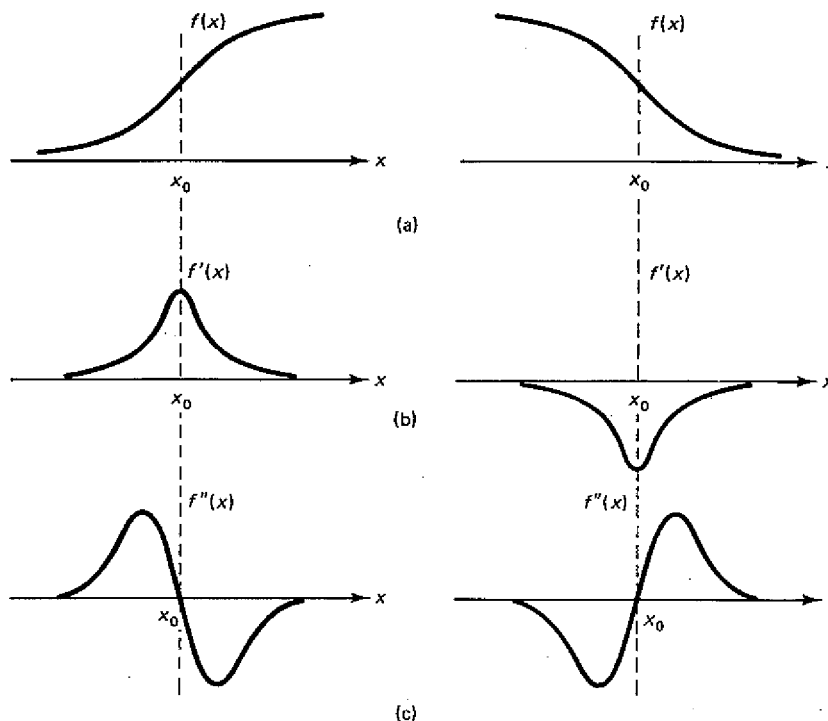


Figure 8.24 (a) $f(x)$, (b) $f'(x)$, and (c) $f''(x)$ for a typical 1-D edge.

checking if $|\nabla f(x, y)|$ is a local maximum in at least one direction. The property that $|\nabla f(x, y)|$ achieves its local maximum in at least one direction is usually checked along a few specified directions. In most cases, it is sufficient to check for local maxima in only the horizontal and vertical directions. If $|\nabla f(x, y)|$ is a local maximum along any one of the specified directions at a potential edge point, the potential edge point is considered to be an edge point. One difficulty with this simple edge thinning algorithm is that it creates a number of minor false edge lines in the vicinity of strong edge lines. One simple method to remove most of these minor false edge lines is to impose the following additional constraints:

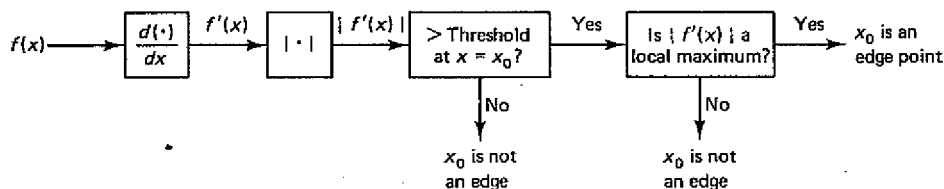


Figure 8.25 System for 1-D edge detection.

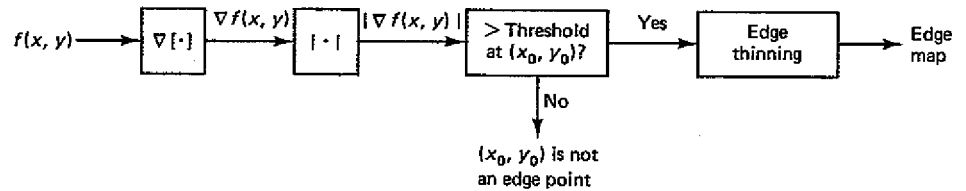


Figure 8.26 System for 2-D edge detection.

- (a) If $|\nabla f(x, y)|$ has a local maximum at (x_0, y_0) in the horizontal direction but not in the vertical direction, (x_0, y_0) is an edge point when
- $$\left| \frac{\partial f(x, y)}{\partial x} \right|_{x=x_0, y=y_0} > k \left| \frac{\partial f(x, y)}{\partial y} \right|_{x=x_0, y=y_0} \quad \text{with } k \text{ typically chosen around } 2.$$
- (b) If $|\nabla f(x, y)|$ has a local maximum at (x_0, y_0) in the vertical direction but not in the horizontal direction, (x_0, y_0) is an edge point when
- $$\left| \frac{\partial f(x, y)}{\partial y} \right|_{x=x_0, y=y_0} > k \left| \frac{\partial f(x, y)}{\partial x} \right|_{x=x_0, y=y_0} \quad \text{with } k \text{ typically chosen around } 2.$$

When $|\nabla f(x, y)|$ has a local maximum at (x_0, y_0) in the horizontal direction but not in the vertical direction, Condition (a) requires that the rate of intensity change along the horizontal direction is significantly larger than that along the vertical direction. Condition (b) is the same as Condition (a) with the roles of x and y reversed. Why these additional constraints remove most of the minor false edges in the vicinity of major edges is discussed in Problem 8.17.

An edge detection system that is based on a function such as $|\nabla f(x, y)|$ is called a *nondirectional edge detector*, since such functions do not have a bias toward any particular direction. If an edge detection system is based on a function that has a bias toward one particular direction, it is called a *directional edge detector*. If we use $|\partial f(x, y)/\partial x|$ instead of $|\nabla f(x, y)|$ in the system in Figure 8.26, for example, the system will detect edges in the vertical direction, but will not respond to edges in the horizontal direction.

For a 2-D sequence $f(n_1, n_2)$, the partial derivatives $\partial f(x, y)/\partial x$ and $\partial f(x, y)/\partial y$ can be replaced by some form of difference. For example, $\partial f(x, y)/\partial x$ may be replaced by

$$\frac{\partial f(x, y)}{\partial x} \leftrightarrow [f(n_1, n_2) - f(n_1 - 1, n_2)]/T, \quad (8.7a)$$

$$[f(n_1 + 1, n_2) - f(n_1, n_2)]/T, \quad (8.7b)$$

$$\text{or} \quad [f(n_1 + 1, n_2) - f(n_1 - 1, n_2)]/(2T). \quad (8.7c)$$

Since the computed derivatives are compared later with a threshold and the threshold can be adjusted, the scaling factors $1/T$ and $1/2T$ can be omitted. Typically the expressions in (8.7) are averaged over several samples to improve the reliability

and continuity of the estimate of $\partial f(x, y)/\partial x$. Examples of "improved" estimates of $\partial f(x, y)/\partial x$ are

$$\frac{\partial f(x, y)}{\partial x} \leftrightarrow [f(n_1 + 1, n_2 + 1) - f(n_1 - 1, n_2 + 1)] + [f(n_1 + 1, n_2) - f(n_1 - 1, n_2)] \\ + [f(n_1 + 1, n_2 - 1) - f(n_1 - 1, n_2 - 1)] \quad (8.8a)$$

or

$$[f(n_1 + 1, n_2 + 1) - f(n_1 - 1, n_2 + 1)] + 2[f(n_1 + 1, n_2) - f(n_1 - 1, n_2)] \\ + [f(n_1 + 1, n_2 - 1) - f(n_1 - 1, n_2 - 1)]. \quad (8.8b)$$

The unnecessary scaling factors have been omitted in (8.8).

The differencing operation in (8.7) and (8.8) can be viewed as the convolution of $f(n_1, n_2)$ with the impulse response of a filter $h(n_1, n_2)$. Examples of impulse responses that can be used in developing directional edge detectors are shown in Figure 8.27. The filters $h(n_1, n_2)$ in Figures 8.27(a) and (b) detect edges in the vertical and horizontal directions and can be viewed as approximating $\partial f(x, y)/\partial x$ and $\partial f(x, y)/\partial y$ respectively. The filters $h(n_1, n_2)$ in Figures 8.27(c) and (d) detect edges along the two diagonal directions. The gradient $\nabla f(x, y)$ in (8.6) can also be expressed in terms of the first-order partial derivatives in a rotated coordinate system. When the rotation is 45 degrees, the directions of partial derivatives are along the two diagonal directions.

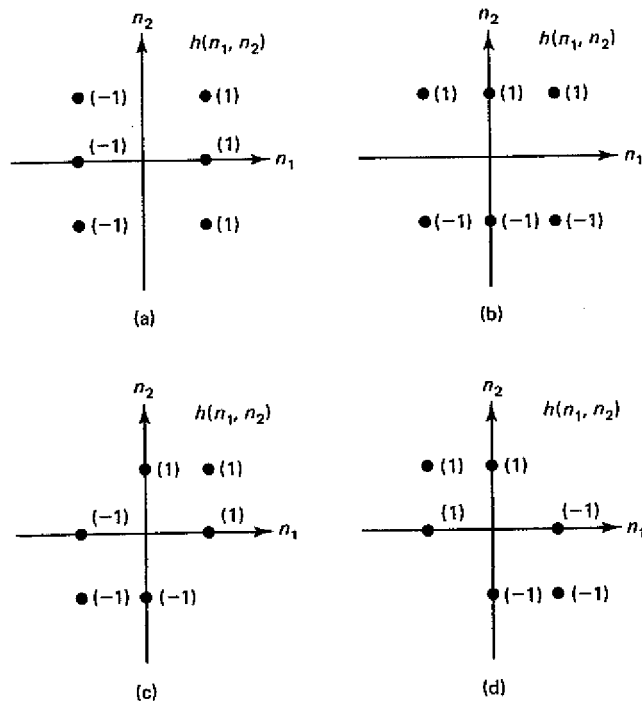


Figure 8.27 Impulse responses of filters that can be used for directional edge detection. (a) Vertical edge detection; (b) horizontal edge detection; (c) and (d) diagonal edge detection.

Nondirectional edge detectors can be developed by discrete approximation of $|\nabla f(x, y)|$ in the system in Figure 8.26. From (8.6),

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (8.9)$$

From (8.9), nondirectional edge detectors can be developed by nonlinear combination of the terms used in the development of directional edge detectors. An example of discrete approximation of (8.9) that can be used for nondirectional edge detectors is given by

$$|\nabla f(x, y)| \rightarrow \sqrt{(f_x(n_1, n_2))^2 + (f_y(n_1, n_2))^2} \quad (8.10)$$

where

$$f_x(n_1, n_2) = f(n_1, n_2) * h_x(n_1, n_2)$$

$$f_y(n_1, n_2) = f(n_1, n_2) * h_y(n_1, n_2)$$

and $h_x(n_1, n_2)$ and $h_y(n_1, n_2)$ are shown in Figure 8.28. The method developed by Sobel [Duda and Hart] is based on (8.10) with $h_x(n_1, n_2)$ and $h_y(n_1, n_2)$ in Figure 8.28. Another example is the method developed by [Roberts], which is based on (8.10) with $h_x(n_1, n_2)$ and $h_y(n_1, n_2)$ shown in Figure 8.29. Depending on exactly how $|\nabla f(x, y)|$ is approximated in the discrete domain, many other variations can be developed.

Figure 8.30 shows the result of edge detection using a directional edge detector. Figure 8.30(a) shows an image of 512×512 pixels. Figure 8.30(b) and (c) show the results of a vertical edge detector and a horizontal edge detector, respectively, applied to the image in Figure 8.30(a). The vertical and horizontal edge detectors are based on $h(n_1, n_2)$ in Figures 8.27(a) and (b). Figures 8.31(a) and (b) show the results of applying the Sobel edge detector and Roberts's edge detector to the image in Figure 8.30(a). Both belong to the class of nondirectional edge detectors, and the specific method of determining the threshold value and checking the local maximum property of an edge used is the same as that used in Figure 8.30.

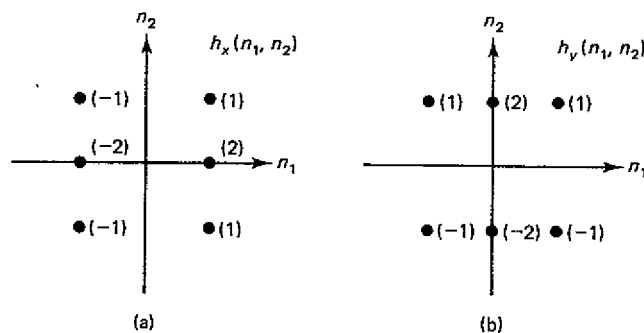


Figure 8.28 Approximation of (a) $\partial f(x, y)/\partial x$ with $f(n_1, n_2) * h_x(n_1, n_2)$; (b) $\partial f(x, y)/\partial y$ with $f(n_1, n_2) * h_y(n_1, n_2)$. Sobel's edge detection method is based on comparison of $\sqrt{(f(n_1, n_2) * h_x(n_1, n_2))^2 + (f(n_1, n_2) * h_y(n_1, n_2))^2}$ with a threshold.

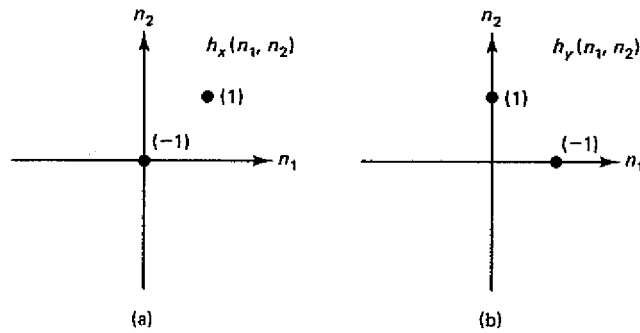


Figure 8.29 Impulse responses of filters used in Roberts's edge detection method. The method is based on comparison of

$$\sqrt{(f(n_1, n_2) * h_x(n_1, n_2))^2 + (f(n_1, n_2) * h_y(n_1, n_2))^2}$$

with a threshold.

There are many variations of the edge detection methods discussed in this section. For example, we could use a different nonlinear combination of $\partial f(x, y)/\partial x$ and $\partial f(x, y)/\partial y$ instead of

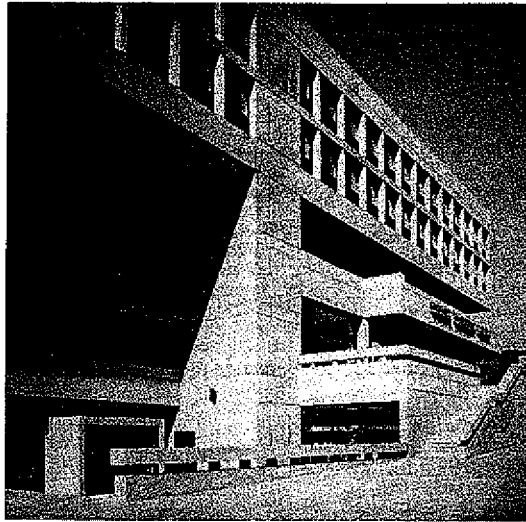
$$\sqrt{(\partial f(x, y)/\partial x)^2 + (\partial f(x, y)/\partial y)^2}$$

in the system in Figure 8.26. Many different methods can also be developed for edge thinning.

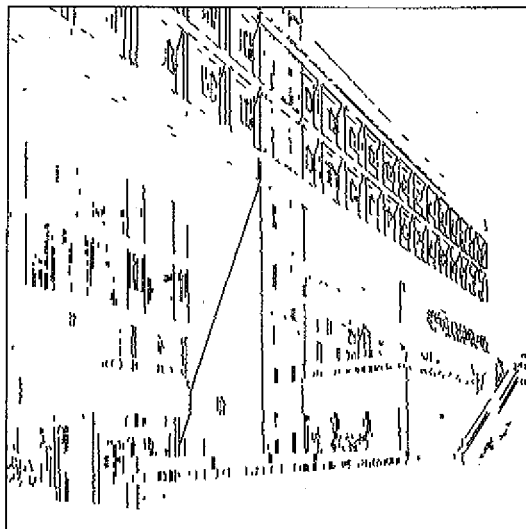
The edge detection methods discussed in this section can be improved in various ways. Methods based on computing some form of gradient or differencing are typically sensitive to noise. A certain number of isolated edge points which appear randomly distributed throughout the edge maps in Figure 8.31 are most likely the result of some background noise or very fine image details. Some noise smoothing using the methods discussed in Section 8.2 or more sophisticated noise reduction methods that we will discuss in Chapter 9 may be desirable prior to applying an edge detection algorithm. Isolated random edge points may also be removed by some simple processing of the edge maps. Gradient-based edge detection methods also cause some discontinuities in the detected edge contours, as can be seen from the edge maps in Figure 8.31. Methods that impose continuity constraints in the detected edge contours can also be developed [Roberts].

8.3.2 Laplacian-Based Methods

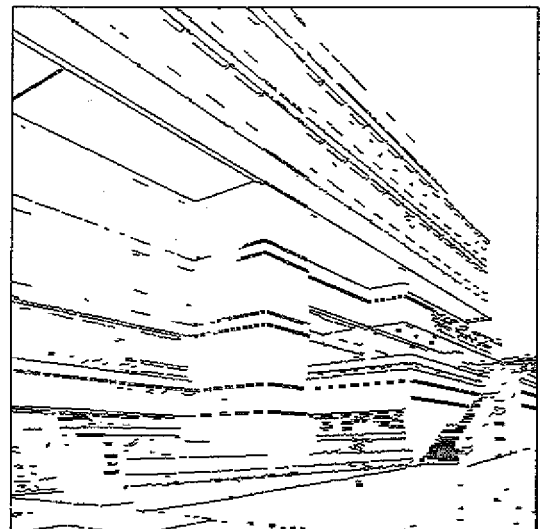
The objective of an edge detection algorithm is to locate the regions where the intensity is changing rapidly. In the case of a 1-D function $f(x)$, searching for regions of rapidly changing intensity corresponds to searching for regions where $f'(x)$ is large. For gradient-based methods, $f'(x)$ is considered large when its magnitude $|f'(x)|$ is greater than a threshold. Another possible way is to assume that $f'(x)$ is large whenever it reaches a local extremum, that is, whenever the second derivative $f''(x)$ has a zero crossing. This is illustrated in Figure 8.24. Declaring zero-crossing points as edges results in a large number of points being declared to be edge points. Since there is no check on the magnitude of $f'(x)$,



(a)



(b)



(c)

Figure 8.30 Edge maps obtained by directional edge detectors. (a) Image of 512×512 pixels; (b) result of applying a vertical edge detector; (c) result of applying a horizontal edge detector.

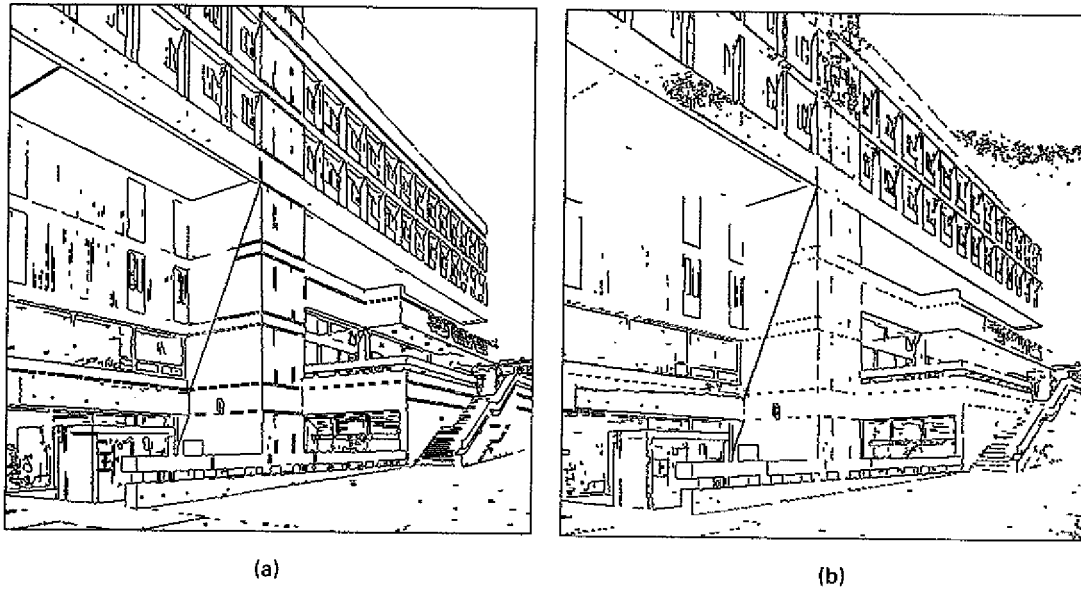


Figure 8.31 Result of applying (a) Sobel edge detector and (b) Roberts's edge detector to the image in Figure 8.30(a).

any small ripple in $f(x)$ is enough to generate an edge point. Due to this sensitivity to noise, the application of a noise reduction system prior to edge detection is very desirable in processing images with background noise.

A generalization of $\partial^2 f(x)/\partial x^2$ to a 2-D function $f(x, y)$ for the purpose of edge detection (see Problem 8.19) is the Laplacian $\nabla^2 f(x, y)$ given by

$$\nabla^2 f(x, y) = \nabla(\nabla f(x, y)) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (8.11)$$

For a 2-D sequence $f(n_1, n_2)$, the partial second derivatives $\partial^2 f(x, y)/\partial x^2$ and $\partial^2 f(x, y)/\partial y^2$ can be replaced by some form of second-order differences. Second-order differences can be represented by convolution of $f(n_1, n_2)$ with the impulse response of a filter $h(n_1, n_2)$. Examples of $h(n_1, n_2)$ that may be used are shown in Figure 8.32. To illustrate that $f(n_1, n_2) * h(n_1, n_2)$ may be viewed as a discrete approximation of $\nabla^2 f(x, y)$, let us consider $h(n_1, n_2)$ in Figure 8.32(a). Suppose we approximate $\partial f(x, y)/\partial x$ by

$$\frac{\partial f(x, y)}{\partial x} \rightarrow f_x(n_1, n_2) = f(n_1 + 1, n_2) - f(n_1, n_2). \quad (8.12)$$

We again omitted the scaling factor, since it does not affect zero-crossing points. Since the forward difference is used in (8.12), we can use the backward difference in approximating $\partial^2 f(x, y)/\partial x^2$:

$$\frac{\partial^2 f(x)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f_x(n_1, n_2) - f_x(n_1 - 1, n_2). \quad (8.13)$$

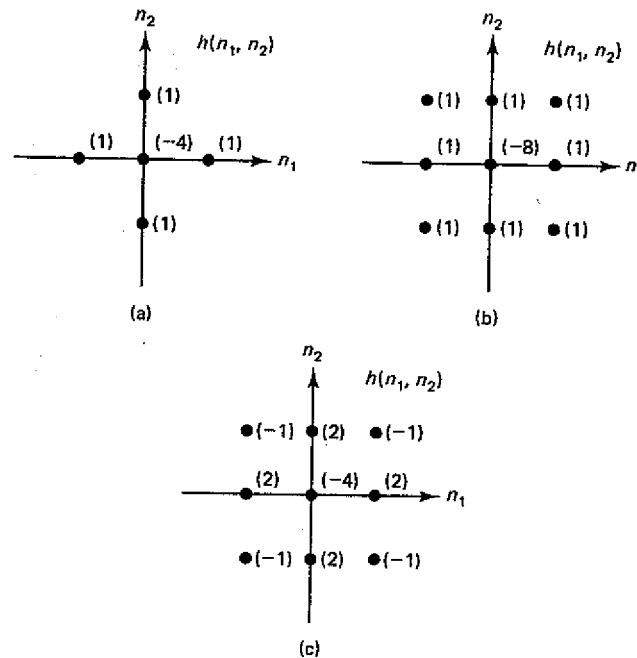


Figure 8.32 Examples of $h(n_1, n_2)$ that may be used in approximating $\nabla^2 f(x, y)$ with $f(n_1, n_2) * h(n_1, n_2)$.

From (8.12) and (8.13),

$$\frac{\partial^2 f(x, y)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f(n_1 + 1, n_2) - 2f(n_1, n_2) + f(n_1 - 1, n_2). \quad (8.14)$$

From (8.11) and (8.14), and approximating $\partial^2 f(x, y)/\partial y^2$ in a similar manner, we obtain

$$\begin{aligned} \nabla^2 f(x, y) &\rightarrow \nabla^2 f(n_1, n_2) = f_{xx}(n_1, n_2) + f_{yy}(n_1, n_2) \\ &= f(n_1 + 1, n_2) + f(n_1 - 1, n_2) + f(n_1, n_2 + 1) \\ &\quad + f(n_1, n_2 - 1) - 4f(n_1, n_2). \end{aligned} \quad (8.15)$$

The resulting $\nabla^2 f(n_1, n_2)$ is $f(n_1, n_2) * h(n_1, n_2)$ with $h(n_1, n_2)$ in Figure 8.32(a). Depending on how the second-order derivatives are approximated, it is possible to derive many other impulse responses $h(n_1, n_2)$, including those shown in Figures 8.32(b) and (c).

Figure 8.33 shows an example where edges were detected by looking for zero-crossing points of $\nabla^2 f(n_1, n_2)$. Figure 8.33(a) shows an image of 512×512 pixels. Figure 8.33(b) shows the zero-crossing points of $\nabla^2 f(n_1, n_2)$, obtained from (8.15) and using the image in Figure 8.33(a) as $f(n_1, n_2)$. Since zero-crossing contours are boundaries between regions, they tend to be continuous lines. As a result, edge thinning necessary in gradient-based methods is not needed in Laplacian-based methods. In addition, algorithms that force continuity in edge contours are not as useful in Laplacian-based methods as in gradient-based methods. As is

clear from Figure 8.33(b), however, choosing all zero-crossing points as edges tends to generate a large number of edge points.

The Laplacian-based methods discussed above generate many “false” edge contours, which typically appear in regions where the local variance of the image is small. As a special case, consider a uniform background region so that $f(n_1, n_2)$ is constant. Since $\nabla^2 f(n_1, n_2)$ is zero and we detect edges from zero-crossing points of $\nabla^2 f(n_1, n_2)$, any small perturbation of $f(n_1, n_2)$ is likely to cause false edge contours. One method to remove many of these false contours is to require that the local variance is sufficiently large at an edge point, as shown in Figure 8.34. The local variance $\sigma_f^2(n_1, n_2)$ can be estimated by

$$\sigma_f^2(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} [f(k_1, k_2) - m_f(k_1, k_2)]^2 \quad (8.16a)$$

$$\text{where} \quad m_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} f(k_1, k_2) \quad (8.16b)$$

with M typically chosen around 2. Since $\sigma_f^2(n_1, n_2)$ is compared with a threshold, the scaling factor $1/(2M+1)^2$ in (8.16a) can be eliminated. In addition, the local variance σ_f^2 needs to be computed only for (n_1, n_2) which are zero-crossing points of $\nabla^2 f(n_1, n_2)$. Figure 8.35 shows the result of applying the system in Figure 8.34 to the image in Figure 8.33(a). Comparison of Figures 8.33(b) and 8.35 shows considerable reduction in the “false” edge contours.

The system in Figure 8.34 can be interpreted as a gradient-based method.

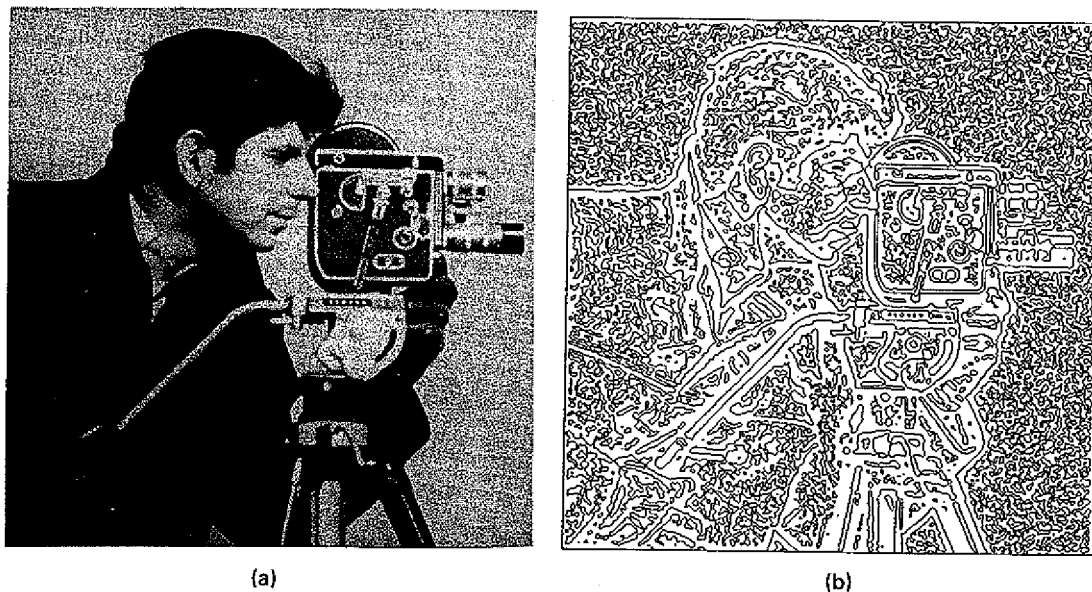


Figure 8.33 Edge map obtained by a Laplacian-based edge detector. (a) Image of 512×512 pixels; (b) result of convolving the image in (a) with $h(n_1, n_2)$ in Figure 8.32(a) and then finding zero-crossing points.

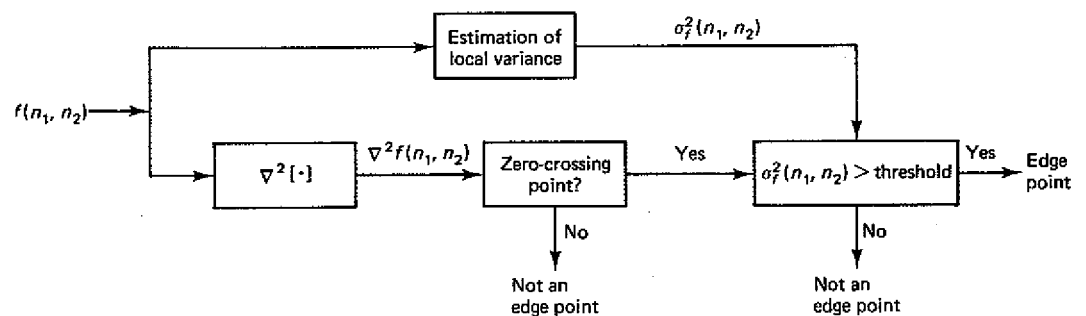


Figure 8.34 Laplacian-based edge detection system that does not produce many false edge contours.

The local variance $\sigma_f^2(n_1, n_2)$ is closely related to the gradient magnitude. Comparing $\sigma_f^2(n_1, n_2)$ with a threshold is similar to comparing the gradient magnitude with a threshold. Requiring that $\nabla^2 f(n_1, n_2)$ crosses zero at an edge can be interpreted as edge thinning. With this interpretation, we can implement the system in Figure 8.34 by computing $\sigma_f^2(n_1, n_2)$ first and then by detecting the zero-crossing points of $\nabla^2 f(n_1, n_2)$ only at those points where $\sigma_f^2(n_1, n_2)$ is above the chosen threshold.

8.3.3 Edge Detection by Marr and Hildreth's Method

In the previous two sections, we discussed edge detection algorithms that produce one edge map from an input image. Marr and Hildreth [Marr and Hildreth; Marr] observed that significant intensity changes occur at different scales (resolution) in an image. For example, blurry shadow regions and sharply focused fine-detail

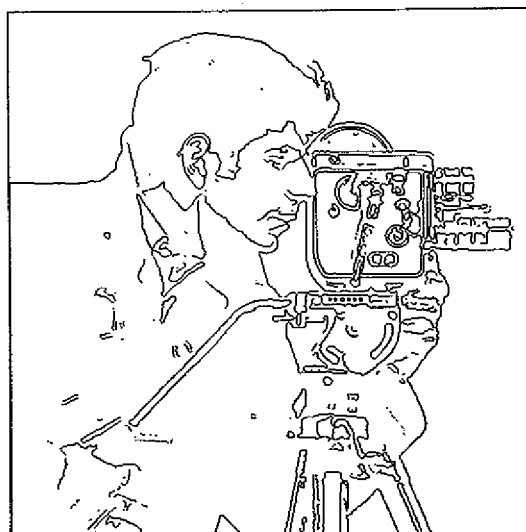


Figure 8.35 Edge map obtained by applying the system in Figure 8.34 to the image in Figure 8.33(a).

regions may be present in the same image. "Optimal" detection of significant intensity changes, therefore, generally requires the use of operators that respond to several different scales. Marr and Hildreth suggested that the original image be band-limited at several different cutoff frequencies and that an edge detection algorithm be applied to each of the images. The resulting edge maps have edges corresponding to different scales.

Marr and Hildreth argue that edge maps of different scales contain important information about physically significant parameters. The visual world is made of elements such as contours, scratches, and shadows, which are highly localized at their own scale. This localization is also reflected in such physically important changes as reflectance change and illumination change. If the same edge is present in a set of edge maps of different scale, it represents the presence of an image intensity change due to a single physical phenomenon. If an edge is present in only one edge map, one reason may be that two independent physical phenomena are operating to produce intensity changes in the same region of the image.

To bandlimit an image at different cutoff frequencies, the impulse response $h(x, y)$ and frequency response $H(\Omega_x, \Omega_y)$ of the lowpass filter proposed [Marr and Hildreth; Canny] is Gaussian-shaped and is given by

$$h(x, y) = e^{-(x^2 + y^2)/(2\pi\sigma^2)} \quad (8.17a)$$

$$H(\Omega_x, \Omega_y) = 2\pi^2\sigma^2 e^{-\pi\sigma^2(\Omega_x^2 + \Omega_y^2)/2} \quad (8.17b)$$

where σ determines the cutoff frequency with larger σ corresponding to lower cutoff frequency. The choice of Gaussian shape is motivated by the fact that it is smooth and localized in both the spatial and frequency domains. A smooth $h(x, y)$ is less likely to introduce any changes that are not present in the original shape. A more localized $h(x, y)$ is less likely to shift the location of edges.

From the smoothed images, edges can be detected by using the edge detection algorithms discussed in the previous two sections. Depending on which method is used, the lowpass filtering operation in (8.17) and the partial derivative operation used for edge detection may be combined. For example, noting that $\nabla^2[\cdot]$ and convolution $*$ are linear, we obtain

$$\begin{aligned} \nabla^2(f(x, y) * h(x, y)) &= f(x, y) * [\nabla^2 h(x, y)] \\ &= f(x, y) * \left[\frac{\partial^2 h(x, y)}{\partial x^2} + \frac{\partial^2 h(x, y)}{\partial y^2} \right] \end{aligned} \quad (8.18)$$

For the Gaussian function $h(x, y)$ in (8.17), $\nabla^2 h(x, y)$ and its Fourier transform are given by

$$\nabla^2 h(x, y) = \frac{e^{-(x^2 + y^2)/(2\pi\sigma^2)}}{(\pi\sigma^2)^2} (x^2 + y^2 - 2\pi\sigma^2) \quad (8.19a)$$

$$F[\nabla^2 h(x, y)] = -2\pi^2\sigma^2 e^{-\pi\sigma^2(\Omega_x^2 + \Omega_y^2)/2} (\Omega_x^2 + \Omega_y^2). \quad (8.19b)$$

Marr and Hildreth chose, for simplicity, to detect edges by looking for zero-crossing points of $\nabla^2 f(x, y)$. Bandlimiting $f(x, y)$ tends to reduce noise, thus reducing the noise sensitivity problem associated with detecting zero-crossing points. The func-

tions $\nabla^2 h(x, y)$ and $-F[\nabla^2 h(x, y)]$ in (8.19) are sketched in Figure 8.36. Clearly, computing $f(x, y) * \nabla^2 h(x, y)$ is equivalent to bandpass filtering $f(x, y)$ where σ^2 in (8.19) is a parameter that controls the bandwidth of the bandpass filter. For a sequence $f(n_1, n_2)$, one approach is to simply replace x and y in (8.19) with n_1 and n_2 .

Figure 8.37 shows an example of the approach under discussion. Figures 8.37(a), (b), and (c) show three images obtained by blurring the original image in Figure 8.33(a) with $h(n_1, n_2)$ obtained by replacing x and y of $h(x, y)$ in (8.17) with n_1 and n_2 with $\sigma^2 = 4, 16$, and 36 , respectively. Figures 8.37(d), (e), and (f) show the images obtained by detecting zero crossings of $f(n_1, n_2) * \nabla^2 h(x, y)|_{x=n_1, y=n_2}$, with $\nabla^2 h(x, y)$ given by (8.19a) for $\sigma^2 = 4, 16$, and 36 , respectively. Marr and Hildreth used the edge maps of different scales, such as those in Figures 8.37(d), (e), and (f) for object representation in their image understanding work.

8.3.4 Edge Detection Based on Signal Modeling

The edge detection algorithms discussed above are general methods, in that they are developed independent of an application context. An alternative approach is to develop an edge detection algorithm specific to a particular application problem. If we know the shape of an edge, for example, this information can be incorporated in the development of an edge detection algorithm. To illustrate how an edge detection algorithm specific to an application problem may be developed, we consider the problem of detecting boundaries of coronary arteries from an angiogram [Abrams].

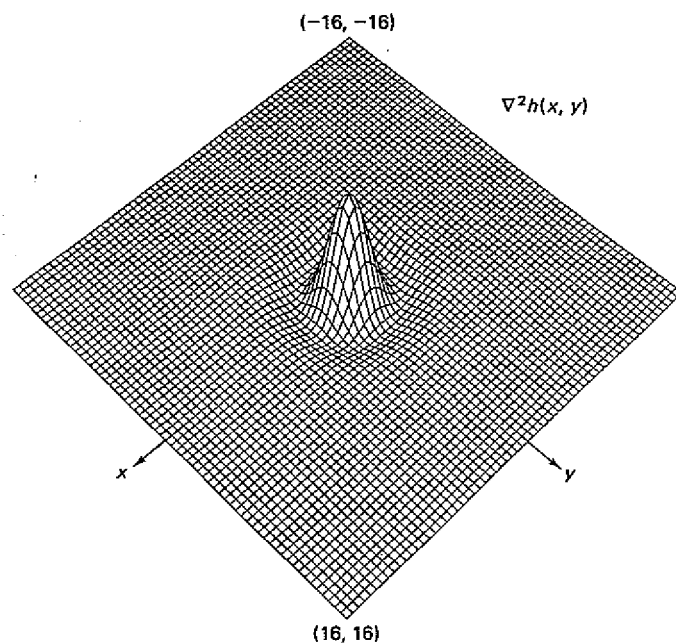
The coronary arteries are the blood vessels which encircle the heart and supply blood to the heart muscle. Narrowing of the coronary arteries prevents adequate blood supply from reaching the heart, causing pain and damage to the heart muscle. Such damage is called coronary disease. To determine the severity of coronary disease, a coronary angiogram is used. An angiogram is an X ray picture of arteries taken after a contrast agent, typically iodine, has been injected into the vessels. The contrast agent is injected directly into the arteries through a catheter in order to achieve high concentrations. An example of a coronary angiogram is shown in Figure 8.38. Different observers making conventional visual evaluations of an angiogram will give widely varying evaluations of the severity of the disease.

The most commonly used measure of an obstruction is percentage of stenosis, which is defined as the maximum percentage of arterial narrowing within a specified length of the vessel. One approach to estimating the percentage of stenosis begins with determining the vessel boundaries from an angiogram. We will be concerned with the problem of detecting the vessel boundaries.

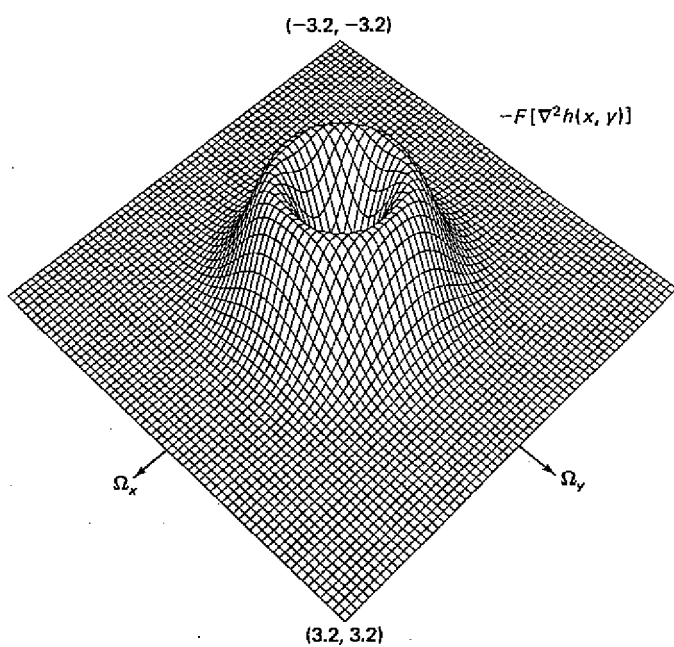
One reasonable model of an angiogram $f(n_1, n_2)$ is given by

$$f(n_1, n_2) = (v(n_1, n_2) + p(n_1, n_2)) * g(n_1, n_2) + w(n_1, n_2) \quad (8.20)$$

where $v(n_1, n_2)$ denotes the vessel, $p(n_1, n_2)$ denotes the background, $g(n_1, n_2)$ denotes blurring, and $w(n_1, n_2)$ denotes the background noise. The vessel function $v(n_1, n_2)$ is derived from a generalized cone model of a 3-D vessel which is con-



(a)



(b)

Figure 8.36 Sketch of (a) $\nabla^2 h(x, y)$ and (b) $-F[\nabla^2 h(x, y)]$ in Equation (8.19) for $\sigma^2 = 1$.

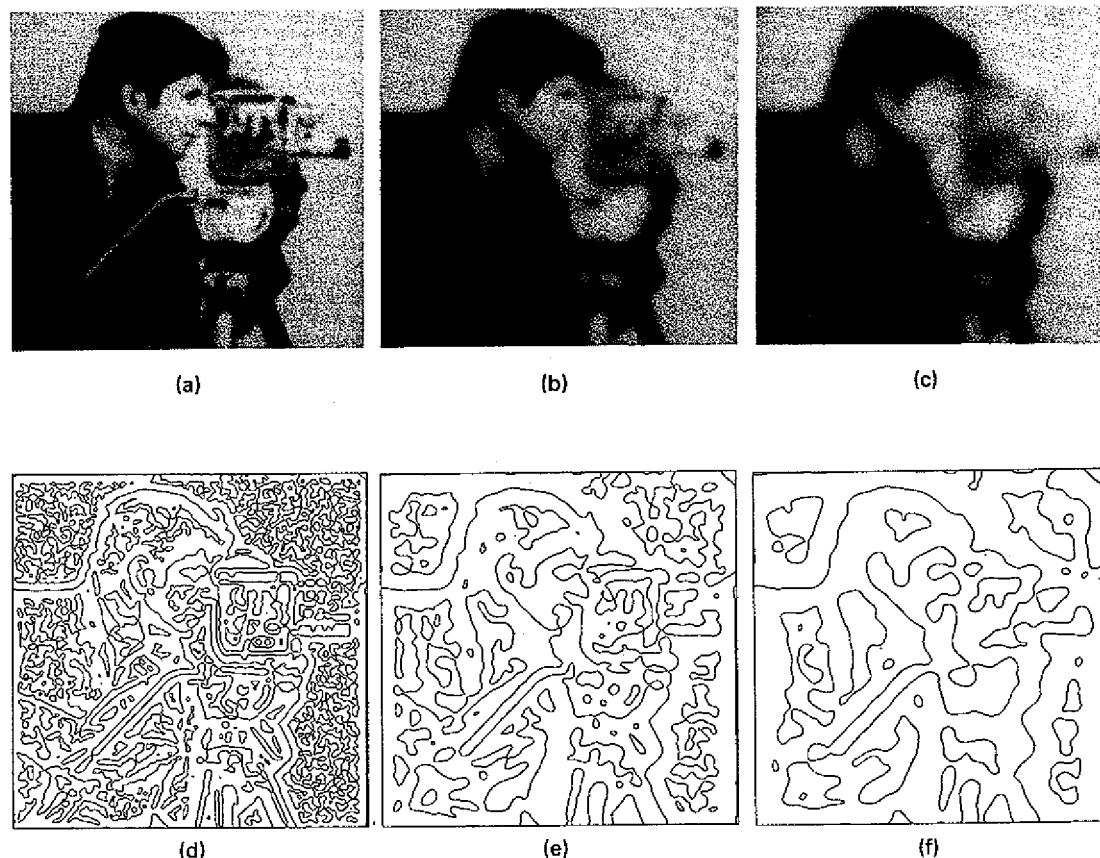


Figure 8.37 Edge maps obtained from lowpass filtered image. Blurred image with (a) $\sigma^2 = 4$; (b) $\sigma^2 = 16$; (c) $\sigma^2 = 36$. Result of applying Laplacian-based algorithm to the blurred image; (d) $\sigma^2 = 4$; (e) $\sigma^2 = 16$; (f) $\sigma^2 = 36$.

tinuous and has elliptical cross sections. The elliptical shape is chosen because of the small number of parameters involved in its characterization and because of some empirical evidence that it leads to a good estimate of percentage of stenosis. The 1-D cross section of $v(n_1, n_2)$, which consists of one blood vessel, is totally specified by three parameters, two representing the blood vessel boundaries and one related to the x-ray attenuation coefficient of iodine. The continuity of the vessel is guaranteed by fitting a cubic spline function to the vessel boundaries. The background $p(n_1, n_2)$ is modeled by a 2-D low-order polynomial. Low-order polynomials are very smooth functions, and their choice is motivated by the observation that objects in the background, such as tissue and bone, are much bigger than the blood vessels. The blurring function $g(n_1, n_2)$ is modeled by a known 2-D Gaussian function that takes into account the blurring introduced at various stages of the imaging process. The noise $w(n_1, n_2)$ is random background noise and is assumed to be white. The parameters in the model of $f(n_1, n_2)$ are the

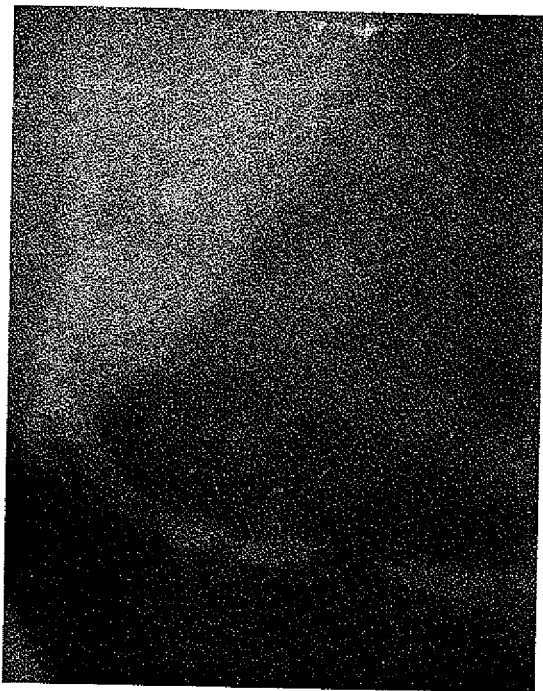
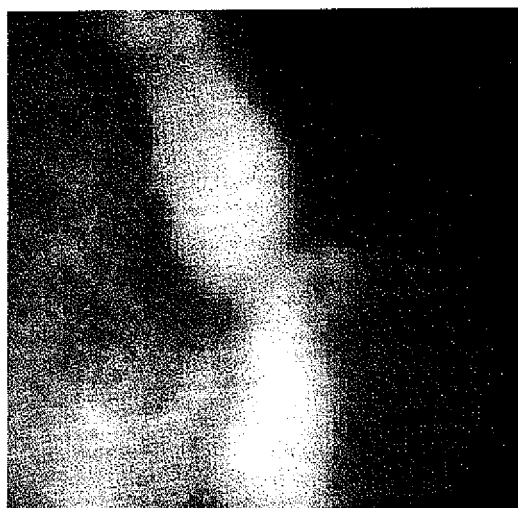


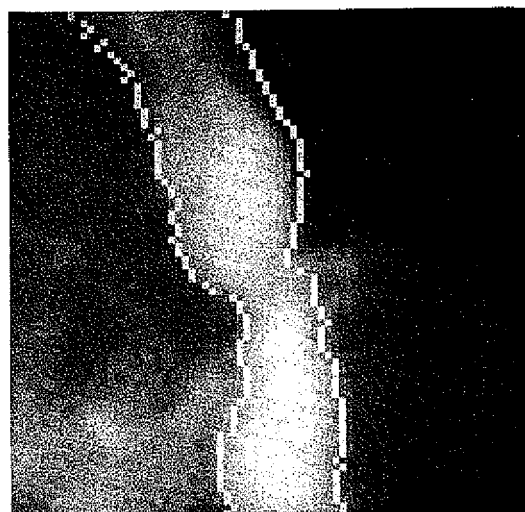
Figure 8.38 Coronary angiogram.

vessel parameters, the polynomial coefficients of $p(n_1, n_2)$, and the noise variance.

The vessels, tissues, bones, and the radiographic imaging process are much more complicated than suggested by the simple model presented above. Nevertheless, the model has been empirically observed to lead to good estimates of the vessel boundaries and corresponding percentage of stenosis. The model parameters may be estimated by a variety of different procedures. One possibility is the maximum likelihood (ML) parameter estimation method discussed in Section 6.1.5. In the ML method, the unknown parameters denoted by θ are estimated by maximizing the probability density function $p_{f(n_1, n_2)|\theta}(f_0(n_1, n_2)|\theta_0)$ where $f(n_1, n_2)$ is the angiogram observation and θ represents all the unknown parameters to be estimated. The ML method applied to vessel boundary detection is a nonlinear problem, but has been solved approximately [Pappas and Lim]. Figures 8.39 and 8.40 illustrate the results of applying the ML parameter estimation method to the detection of the blood vessels using the 1-D version of the 2-D model in (8.20). In the 1-D version, $f(n_1, n_2)$ in (8.20) is considered a 1-D sequence with variable n_1 for each n_2 . Computations simplify considerably when the 1-D model is used. Figure 8.39(a) shows the original angiogram of 80×80 pixels, and Figure 8.39(b) shows the detected vessel boundaries superimposed on the original image. Figure 8.40 is another example. Developing an edge detection algorithm specific to an application problem is considerably more complicated than applying the general edge detection algorithms discussed in previous sections. However, it has the potential of leading to much more accurate edge detection.

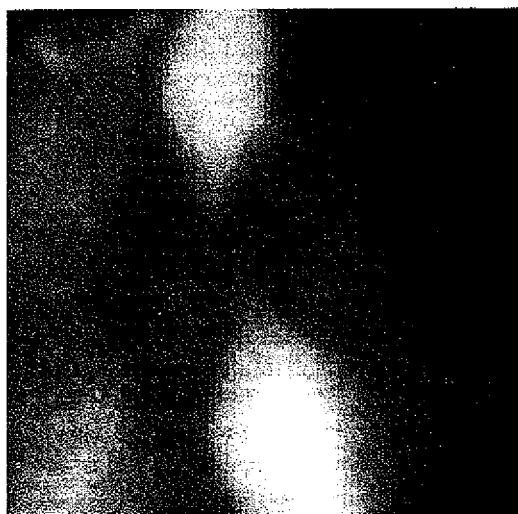


(a)

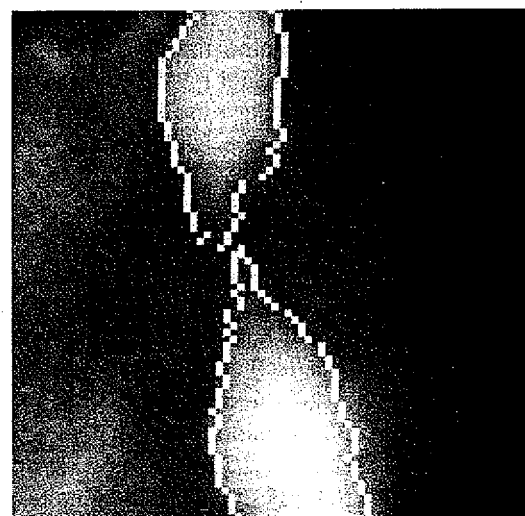


(b)

Figure 8.39 Example of vessel boundary detection from an angiogram by signal modeling. (a) Angiogram of 80×80 pixels; (b) vessel boundary detection.



(a)



(b)

Figure 8.40 Another example of vessel boundary detection from an angiogram by signal modeling. (a) Angiogram of 80×80 pixels; (b) vessel boundary detection.

Appendix

Magnetic Recording as a Communications Channel

Jack Keil Wolf

Center for Magnetic Recording Research, University of California, San Diego, La Jolla, CA 92093-0401, USA

A communications channel is defined by specifying its input, its output and the probabilistic rules by which the allowable inputs produce outputs. An often used, but overly simplistic, model for a digital magnetic recording channel is to describe the input as a continuous waveform that takes on only one of two values (say $+A$ and $-A$) and to describe the output as the sum of a linearly filtered version of the input and independent Gaussian noise.

One aim of this talk is to explain how this simple model need be modified to take into account some of the important phenomena that occurs for high density recording. A second aim of this talk is to describe some of the coding schemes that are presently being used (or are about to be used) in digital recording systems.

My grandchildren sometimes are rewarded with "special treats" when they are especially good. Your special treat for coming at this early hour on the last day of the conference (a schedule which was chosen by my good friend, the program committee chairman) will be to hear music played by the various digital recording formats available today: i.e., CD, DCC, DAT and Minidisk. (This assumes that the necessary audio equipment will be available.)

Signal-to-Noise Ratio Definition for Magnetic Recording Channels with Transition Noise

Jaekyun Moon, *Senior Member, IEEE*

Abstract—This paper proposes a new signal-to-noise ratio (SNR) definition for magnetic recording channels with both additive and medium noise components. The proposed SNR is a generalized version of E_b/N_o , the information bit energy to noise spectral height ratio, widely used in average-power-constrained communication channels with additive white noise. The goal is to facilitate comparison of efficiencies of read channels that may operate at different symbol densities because of varying code rates.

Index Terms—Bit error rate, code rate loss, medium noise, signal-to-noise ratio, transition noise.

I. INTRODUCTION

We wish to define the signal-to-noise ratio (SNR) in such a way that we can compare efficiencies of different codes/detectors running at different symbol densities based on the SNR required to meet a fixed bit error rate (BER) target. As such, the SNR definition should be free of the symbol density. However, the statistical properties of medium noise depend on the written pattern as well as on the symbol (or written bit) density, and defining an SNR to meet this goal is not a trivial matter.

Let us first review SNR definitions for additive noise channels. In communication channels with additive white noise (AWN) and an average transmitter power constraint, the E_b/N_o ratio, the information bit energy to noise spectral height ratio, serves this purpose. It is well known that the error probability of such systems is mainly determined by the ratio

$$\beta_{\text{com}} = \frac{E_s}{N_o} = \frac{rE_b}{N_o} = \frac{P_s}{N_o/T}, \quad (1)$$

where E_s is the symbol energy, r is the code rate, and P_s denotes the average transmitter power. This ratio is sometimes referred to as the detection SNR. The penalty associated with the code rate loss is apparent in (1).

For magnetic recording channels with AWN, E_t/N_o , the isolated transition pulse energy to noise spectral height ratio, can serve this function as discussed in [1]. A magnetic recording channel can be viewed as the “ $h(t)$ -constrained” channel, where $h(t)$ is the channel’s response to a single transition; there is no average or peak power constraint here, but for a given head/medium interface $h(t)$ is fixed, and coding and signal processing engineers must live with it. At low-to-medium recording densities, where single errors are more likely than

multiple errors, the detection SNR of an optimal sequence detector is given by

$$\beta_{\text{mag}} = \frac{1}{2} \frac{\int_{-\infty}^{\infty} [h(t) - h(t-T)]^2 dt}{N_o/2T} = \frac{E_d}{N_o}, \quad (2)$$

where T denotes the symbol period and E_d is the dibit energy or the energy associated with a single nonreturn-to-zero (NRZ) written bit. This ratio can be interpreted as the ratio of the isolated dibit power to the in-band noise power (i.e., the noise power within the $|f| \leq 1/2T$ band). It can be shown that E_d decreases more rapidly than E_s of an AWN, average power-constrained channel, as the code rate decreases. This means that the penalty associated with a code rate loss is more severe with the magnetic recording channel than is an AWN channel with an average power constraint. For the Lorentzian channel, where

$$h(t) = \frac{V_o}{1 + (2t/PW50)^2} = \sqrt{\frac{4E_t}{\pi PW50}} \cdot \frac{1}{1 + (2t/PW50)^2} \quad (3)$$

(2) reduces to

$$\beta_{\text{mag}} = \frac{2}{D_s^2 + 1} \cdot \frac{E_t}{N_o} = \frac{2r^2}{D_u^2 + r^2} \cdot \frac{E_t}{N_o}, \quad (4)$$

where D_s denotes the symbol density defined as $PW50/T$ and D_u is the user density given by $r \cdot D_s$. It can be seen that β_{mag} is roughly proportional to r^2 (assuming D_u^2 is considerably larger than r^2), as noted in [1],¹ whereas β_{com} is proportional to r , confirming a more severe code rate penalty in magnetic recording than in power constrained communication channels.

II. MAGNETIC RECORDING CHANNELS WITH TRANSITION NOISE

The E_t/N_o definition, however, cannot be used when transition noise exists. We propose the following SNR definition to handle mixed noise channels:

$$\text{SNR} = \frac{E_t}{N_0 + M_0} = \frac{E_t}{N_\alpha}, \quad (5)$$

where M_0 is twice the average energy² of the medium noise associated with each transition, $N_\alpha = N_0 + M_0$, and α denotes

¹Ryan used the notation E_t to denote the energy in the isolated transition response, but we use the subscript t to emphasize it is the energy in the isolated transition response rather than isolated dibit response. Also, note that $h(t)$ here is the response to a full magnetization step, whereas the transition response in [1] is the half magnetization step, leading to the relationship: $E_t = 4E_i$.

²It is safe to assume that the medium noise voltage associated with each transition is square integrable.

the medium noise power expressed as a percentage of the total in-band noise power, i.e.,

$$\alpha = \frac{M_0}{N_0 + M_0} \times 100. \quad (6)$$

We can further write $N_0 = [(100 - \alpha)/100]N_\alpha$ and $M_0 = (\alpha/100)N_\alpha$ with $0 \leq \alpha \leq 100$. With $\alpha = 0$, the SNR reduces to E_t/N_0 . Letting $n_m(t)$ denote the medium noise voltage waveform associated with a given transition, the average medium noise energy per transition can be defined as

$$M_0/2 = \int_{-\infty}^{\infty} \overline{n_m^2(t)} dt, \quad (7)$$

where the statistical average is taken over transitions written at different positions in the disk. If the medium noise is statistically independent among transitions (as in the case of disks exhibiting a linear increase of noise power versus linear density), the total medium noise power originating from the all-transitions (or 1T) pattern is easily shown to be $\int_{-\infty}^{\infty} \overline{n_m^2(t)} dt/T$. This means that

$$\frac{M_0}{2T} = (\text{total medium noise power in 1T pattern}) \quad (8)$$

and M_0 can be viewed as the equivalent (single-sided) band-limited white noise spectrum of the medium noise for this pattern (i.e., M_0 is the height of a box-car power spectral density (PSD) of width $1/2T$ that integrates to the same value as the 1T pattern's medium noise PSD).

The SNR definition of (5) can be rewritten as

$$\text{SNR} = \frac{E_t}{N_0 + M_0} = \frac{1}{2} \cdot \frac{E_t/T}{\frac{N_0}{2T} + \frac{M_0}{2T}}, \quad (9)$$

giving rise to an interpretation

$$\text{SNR} = \frac{1}{2} \cdot \frac{E_t/T}{(\text{total inband noise power with 1T pattern})}, \quad (10)$$

which can be viewed as the ratio of the isolated transition pulse power to the overall in-band noise power.

For the first-order position jitter and width variation (1PW) noise model [2], [3], M_0 can be written as

$$M_0 = 2\sigma_j^2 I_t + 2\sigma_w^2 I_w, \quad (11)$$

where σ_j^2 and σ_w^2 are the variances of position jitter and width variation, respectively, and

$$I_t = \int_{-\infty}^{\infty} [\partial h / \partial t]^2 dt, \quad (12)$$

$$I_w = \int_{-\infty}^{\infty} [\partial h / \partial w]^2 dt. \quad (13)$$

With the above definitions,³ specifying E_t/N_α with a particular value of α determines both N_0 and M_0 [assuming $h(t)$ is given as well]. Once M_0 is determined, the noise parameters necessary for BER simulation or analysis are specified. For example, with the 1PW model, both σ_j^2 and σ_w^2 are specified once

³Here, we are implying $h(t)$ should really be written as $h(t, w)$, where w is the width of the magnetization profile.

M_0 is known, provided the ratio of position jitter noise power to the width variation noise power is also known. Once N_α , σ_j^2 , and σ_w^2 as well as the symbol period are specified, we can proceed with performance analysis or BER simulations to compare codes/detectors operating at different symbol or user densities. With the microtrack noise model [4], fixing M_0 will determine the number of microtracks and the magnetization width parameter, which are required to simulate the recording channel output. Thus, assuming $h(t)$ and the noise model are predetermined, specifying E_t/N_α with a particular value of α completely characterizes the channel for analysis and simulation purposes.

Note that the key assumption here is that M_0 is independent of the symbol period T so that the noise composition parameter α is also independent of T . This means that the E_t/N_α definition is free of symbol density, as desired. This assumption will hold for any recording systems in which the total integrated medium noise power increases linearly with $1/T$ (i.e., where the intrinsic medium noise power per transition remains the same as the symbol pattern or density changes). This condition is met with well-designed media that do not exhibit supralinear noise increase with density (see, for example, [5]).

For the Lorentzian channel, it can be shown that⁴

$$I_t = I_w = V_o^2 \cdot \pi \cdot \frac{1}{2PW50} = \frac{2E_t}{PW50^2} \quad (14)$$

and that

$$E_t/N_\alpha = \frac{E_t}{N_\alpha + 4 \frac{(\sigma_j^2 + \sigma_w^2)}{PW50^2} E_t} = \frac{E_t/N_o}{1 + 4 \frac{(\sigma_j^2 + \sigma_w^2)}{PW50^2} \frac{E_t}{N_o}}. \quad (15)$$

As a point of reference, to meet a 10^{-5} target BER for the 1PW Lorentzian channel at user density 2.5, EPR4ML (extended partial response class 4 maximum likelihood) with a rate 16/17 runlength-limited (RLL) code requires an E_t/N_o slightly above 24 dB and an E_t/N_{90} approximately equal to 23 dB with medium noise comprising only position (independent) jitter [6].

III. RELATIONSHIPS WITH OTHER SNR DEFINITIONS

Let us compare the proposed E_t/N_α ratio with the SNR definition recording engineers are more familiar with, namely, the peak isolated pulse amplitude to the total in-band noise ratio for a pattern written at some particular density, say, D_s . This SNR can be expressed as

$$\text{SNR}_p(D_s, \gamma) = \frac{V_o^2}{NP(D_s, \gamma)}, \quad (16)$$

where V_o is the peak-zero amplitude of $h(t)$, $NP(D_s, \gamma)$ is the total $1/2T$ -band noise power measured with a data pattern whose symbol density is D_s , and average transition density (total number of transitions divided by the total number of written bits) is γ . We can further write

$$\text{SNR}_p(D_s, \gamma) = \frac{V_o^2}{N_0 \frac{\gamma M_0}{2T} + \frac{\gamma M_0}{2T}} = \frac{8}{\pi \cdot D_s} \cdot \frac{E_t}{N_0 + \gamma M_0}, \quad (17)$$

⁴For the computation of I_w , we assume $w = PW50/2$ here.

where the last equality holds for the Lorentzian pulse. Comparing (17) and (5), the two SNRs can be related via

$$\text{SNR}_p(D_s, \gamma = 1) = \frac{8}{\pi \cdot D_s} \cdot \frac{E_t}{N_\alpha}. \quad (18)$$

Another SNR definition of interest is the mean-square (ms) SNR ratio defined as

$$\text{SNR}_{\text{ms}}(D_s, \gamma) = \frac{\frac{1}{2\Delta} \int_{-\Delta}^{\Delta} s^2(t) dt}{\frac{1}{2\Delta} \int_{-\Delta}^{\Delta} n^2(t) dt}, \quad (19)$$

where Δ is some large interval ($\Delta \gg T$), and $s(t)$ and $n(t)$ are overall signal and noise (band-limited to $1/2T$) voltages, respectively, corresponding to a written pattern with symbol density D_s and average transition density γ . For Lorentzian, we have

$$\text{SNR}_{\text{ms}}(D_s, \gamma) = \frac{E_t}{(D_s^2 + 1)(\gamma M_0 + N_0)}. \quad (20)$$

This SNR can be relate to E_t/N_α via

$$\text{SNR}_{\text{ms}}(D_s, \gamma = 1) = \frac{1}{D_s^2 + 1} \cdot \frac{E_t}{N_\alpha}. \quad (21)$$

IV. CONCLUSIONS

A new SNR definition has been proposed that can be used to compare read channels with different code rates. It can handle channels with medium noise. The proposed SNR definition, denoted by E_t/N_α with α representing the percentage medium noise power, is applicable to any medium noise model as long as the model yields a linear relationship between the total medium noise power and the number of transitions in a given length of a track. The relationships between E_t/N_α and other often-used SNR ratios, namely, peak-zero SNR and ms SNR, have been established.

REFERENCES

- [1] W. Ryan, "Optimal code rates for concatenated codes on a PR4-equalized magnetic recording channel," in *Proc. CISS'99*.
- [2] J. Moon, L. R. Carley, and R. R. Katti, "Density dependence of noise in thin metallic longitudinal media," *J. Appl. Phys.*, vol. 63, pp. 3254–3256, Apr. 1988.
- [3] J. Moon, "Discrete-time modeling of transition-noise-dominant channels and study of detection performance," *IEEE Trans. Magn.*, vol. 27, pp. 4573–4578, Nov. 1991.
- [4] J. Caroselli and J. K. Wolf, "Application of a new simulation model for media noise limited magnetic recording channels," *IEEE Trans. Magn.*, vol. 32, pp. 3917–3919, Sept. 1996.
- [5] E. Yen *et al.*, "Case study of media noise mechanisms in longitudinal recording," *IEEE Trans. Magn.*, vol. 35, pp. 2730–2732, Sept. 1999.
- [6] T. Oenning and J. Moon, "Modeling the Lorentzian magnetic recording channels with transition noise," *IEEE Trans. Magn.*, to be published.

Phase Congruency Detects Corners and Edges

Peter Kovesi

School of Computer Science & Software Engineering
The University of Western Australia
Crawley, W.A. 6009
pk@csse.uwa.edu.au

Abstract. There are many applications such as stereo matching, motion tracking and image registration that require so called ‘corners’ to be detected across image sequences in a reliable manner. The Harris corner detector is widely used for this purpose. However, the response from the Harris operator, and other corner operators, varies considerably with image contrast. This makes the setting of thresholds that are appropriate for extended image sequences difficult, if not impossible. This paper describes a new corner and edge detector developed from the phase congruency model of feature detection. The new operator uses the principal moments of the phase congruency information to determine corner and edge information. The resulting corner and edge operator is highly localized and has responses that are invariant to image contrast. This results in reliable feature detection under varying illumination conditions with fixed thresholds. An additional feature of the operator is that the corner map is a strict subset of the edge map. This facilitates the cooperative use of corner and edge information.

1 Introduction

With the impressive reconstruction results that have been achieved by those working in projective geometry (see for example Hartley and Zisserman [1]) there has been a renewed interest in the detection of so called ‘corners’, or ‘interest points’. The success of these reconstructions depend very much on the reliable and accurate detection of these points across image sequences.

The definition of a corner is typically taken to be a location in the image where the local autocorrelation function has a distinct peak. A variety of operators have been devised to detect corners. These include those developed by Moravec [2], Harris and Stephens [3], Beaudet [4], Kitchen and Rosenfeld [5], and Cooper et al. [6]. Corner detectors based on the local energy model of feature perception have been developed by Rosenthaler et al. [7], and Robbins and Owens [8]. More recently the SUSAN operator has been proposed by Smith and Brady [9]. Of these the Harris operator probably remains the most widely used.

A common problem with all these operators, except the SUSAN operator, is that the corner response varies considerably with image contrast. This makes the setting of thresholds difficult. Typically we are interested in tracking features over increasingly extended image sequences. The longer the sequence the

greater the variations in illumination conditions one can expect, and the setting of appropriate thresholds becomes increasingly difficult, if not impossible. Inevitably thresholds have to be set at levels lower than ideal because detecting too many features is a lesser evil than detecting not enough. Stereo and motion reconstruction algorithms are then faced with the problem of dealing with very large clouds of noisy corner points, often greatly compromising their operation. Matching operations, which usually have rapidly increasing running time as a function of input size, suffer greatly in these conditions. Typically, considerable effort has to be devoted to the cleaning up of the output of the corner detector and to the elimination of outliers. The success of this cleaning up and outlier elimination process is usually crucial to the success of the reconstruction algorithm. Indeed, a very significant proportion of Hartley and Zisserman's book [1] is devoted to robust estimation techniques.

Another difficulty many of these operators have is that the Gaussian smoothing that is employed to reduce the influence of noise can corrupt the location of corners, sometimes considerably. The SUSAN operator deserves some special comment here because it does not suffer from these problems outlined above. It identifies features by determining what fraction of a circular mask has values the same, or similar, to the value at the centre point. Thresholds are therefore defined in terms of the size of the mask and no image smoothing is required. However, the SUSAN operator assumes that edges and corners are formed by the junctions of regions having constant, or near constant, intensity, and this limits the junction types that can be modeled.

To address the many problems outlined above this paper describes a new corner and edge detector developed from the phase congruency model of feature detection. The new operator uses the principal moments of the phase congruency information to determine corner and edge information. Phase congruency is a dimensionless quantity and provides information that is invariant to image contrast. This allows the magnitudes of the principal moments of phase congruency to be used directly to determine the edge and corner strength. The minimum and maximum moments provide feature information in their own right; one does not have to look at their ratios. If the maximum moment of phase congruency at a point is large then that point should be marked as an edge. If the minimum moment of phase congruency is also large then that point should also be marked as a 'corner'. The hypothesis being that a large minimum moment of phase congruency indicates there is significant phase congruency in more than one orientation, making it a corner.

The resulting corner and edge operator is highly localized and the invariance of the response to image contrast results in reliable feature detection under varying illumination conditions with fixed thresholds. An additional feature of the operator is that the corner map is a strict subset of the edge map. This facilitates the cooperative use of corner and edge information.

This paper is organized as follows: first the phase congruency model of feature perception is reviewed. We then examine how the phase congruency responses over several orientations can be analyzed in terms of moments to provide both

edge and corner information. Finally the performance is assessed relative to the commonly used Harris operator.

2 The Phase Congruency Model of Feature Detection

Rather than assume a feature is a point of maximal intensity gradient, the *Local Energy Model* postulates that features are perceived at points in an image where the Fourier components are maximally in phase as shown in Figure 1 [10]. Notice

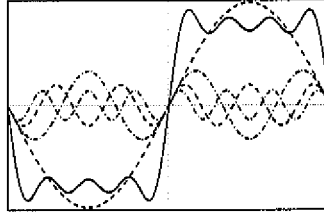


Fig. 1. Fourier series of a square wave and the sum of the first four terms.

how the Fourier components are all *in phase* at the point of the step in the square wave. Congruency of phase at *any angle* produces a clearly perceived feature [11]. The angle at which the congruency occurs dictates the feature type, for example, step or delta.

The Local Energy Model model was developed by Morrone et al. [10] and Morrone and Owens [12]. Other work on this model of feature perception can be found in Morrone and Burr [13], Owens et al. [14], Venkatesh and Owens [15], and Kovesi [16–20, 11]. The work of Morrone and Burr [13] has shown that this model successfully explains a number of psychophysical effects in human feature perception.

The measurement of phase congruency at a point in a signal can be seen geometrically in Figure 2. The local, complex valued, Fourier components at a location x in the signal will each have an amplitude $A_n(x)$ and a phase angle $\phi_n(x)$. Figure 2 plots these local Fourier components as complex vectors adding head to tail. The magnitude of the vector from the origin to the end point is the *Local Energy*, $|E(x)|$.

The measure of phase congruency developed by Morrone et al. [10] is

$$PC_1(x) = \frac{|E(x)|}{\sum_n A_n(x)}. \quad (1)$$

Under this definition phase congruency is the ratio of $|E(x)|$ to the overall path length taken by the local Fourier components in reaching the end point. If all the Fourier components are in phase all the complex vectors would be aligned and the ratio of $|E(x)|/\sum_n A_n(x)$ would be 1. If there is no coherence of phase

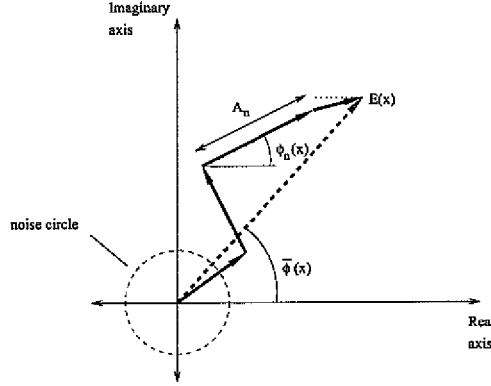


Fig. 2. Polar diagram showing the Fourier components at a location in the signal plotted head to tail. The weighted mean phase angle is given by $\bar{\phi}(x)$. The noise circle represents the level of $E(x)$ one can expect just from the noise in the signal.

the ratio falls to a minimum of 0. Phase congruency provides a measure that is independent of the overall magnitude of the signal making it invariant to variations in image illumination and/or contrast. Fixed threshold values of feature significance can then be used over wide classes of images.

It can be shown that this measure of phase congruency is a function of the cosine of the deviation of each phase component from the mean

$$PC_1(x) = \frac{\sum_n A_n (\cos(\phi(x) - \bar{\phi}(x)))}{\sum_n A_n(x)}. \quad (2)$$

This measure of phase congruency does not provide good localization and it is also sensitive to noise. Kovessi [18, 19] developed a modified measure consisting of the cosine minus the magnitude of the sine of the phase deviation; this produces a more localized response. This new measure also incorporates noise compensation:

$$PC_2(x) = \frac{\sum_n W(x) [A_n(x) (\cos(\phi_n(x) - \bar{\phi}(x)) - |\sin(\phi_n(x) - \bar{\phi}(x))|) - T]}{\sum_n A_n(x) + \epsilon}. \quad (3)$$

The term $W(x)$ is a factor that weights for frequency spread (congruency over many frequencies is more significant than congruency over a few frequencies). A small constant, ϵ is incorporated to avoid division by zero. Only energy values that exceed T , the estimated noise influence, are counted in the result. The symbols $[\]$ denote that the enclosed quantity is equal to itself when its value is positive, and zero otherwise. In practice local frequency information is obtained via banks of Gabor wavelets tuned to different spatial frequencies, rather than via the Fourier transform. The appropriate noise threshold, T is readily determined from the statistics of the filter responses to the image. For details of this phase congruency measure and its implementation see Kovessi [19–21].

3 Combining Phase Congruency Information over many Orientations

A weakness of previous implementations of phase congruency has been the way in which information over many orientations is used and combined. The definition of phase congruency outlined above only applies to 1D signals. To obtain an overall measure of phase congruency in 2D local energy is first calculated in several orientations, typically six, using data from oriented 2D Gabor wavelets. Equation 3 is modified so that the numerator is the weighted and noise compensated local energy summed over all orientations, and the denominator is the total sum of filter response amplitudes over all orientations and scales. While this approach produces a phase congruency measure that results in a very good edge map it ignores information about the way phase congruency varies with orientation at each point in the image.

To include information about the way phase congruency varies with orientation we can proceed as follows: calculate phase congruency *independently* in each orientation using equation 3, compute moments of phase congruency and look at the variation of the moments with orientation. The principal axis, corresponding to the axis about which the moment is minimized, provides an indication of the orientation of the feature. The magnitude of the maximum moment, corresponding to the moment about an axis perpendicular to the principal axis, gives an indication of the significance of the feature. If the minimum moment is also large we have an indication that the feature point has a strong 2D component to it, and should therefore be additionally classified as a 'corner'.

Following the classical moment analysis equations [22] we compute the following at each point in the image:

$$a = \sum (PC(\theta) \cos(\theta))^2 \quad (4)$$

$$b = 2 \sum (PC(\theta) \cos(\theta)) \cdot (PC(\theta) \sin(\theta)) \quad (5)$$

$$c = \sum (PC(\theta) \sin(\theta))^2, \quad (6)$$

where $PC(\theta)$ refers to the phase congruency value determined at orientation θ , and the sum is performed over the discrete set of orientations used (typically six). The angle of the principal axis Φ is given by

$$\Phi = \frac{1}{2} \text{atan2} \left(\frac{b}{\sqrt{b^2 + (a - c)^2}}, \frac{a - c}{\sqrt{b^2 + (a - c)^2}} \right). \quad (7)$$

The maximum and minimum moments, M and m respectively, are given by

$$M = \frac{1}{2}(c + a + \sqrt{b^2 + (a - c)^2}) \quad (8)$$

$$m = \frac{1}{2}(c + a - \sqrt{b^2 + (a - c)^2}). \quad (9)$$

This calculation of the maximum and minimum moments, along with the principal axis, corresponds to performing a singular value decomposition on a phase congruency covariance matrix. The moments correspond to the singular values.

3.1 Comparison with the Harris Operator

A comparison with the Harris operator is appropriate given its widespread use. The analysis described above is similar to that adopted by Harris and Stephens [3]. However they consider the minimum and maximum eigenvalues, α and β , of the image *gradient* covariance matrix in developing their corner detector. The gradient covariance matrix is given by

$$G = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (10)$$

where I_x and I_y denote the image gradients in the x and y directions. A ‘corner’ is said to occur when the two eigenvalues are large and similar in magnitude. To avoid an explicit eigenvalue decomposition Harris and Stephens devise a measure using the determinant and trace of the gradient covariance matrix

$$R = \det(G) - k(\text{tr}(G))^2, \quad (11)$$

where $\det(G) = \alpha\beta$ and $\text{tr}(G) = \alpha + \beta$, the parameter k is traditionally set to 0.04. This produces a measure that is large when both α and β are large. However we have the problem of determining what is large. Noting that elements of the image gradient covariance matrix have units of intensity gradient squared we can see that the determinant, and hence the measure R will have units of intensity gradient to the fourth. This explains why the Harris operator is highly sensitive to image contrast variations which, in turn, makes the setting of thresholds exceedingly difficult. Some kind of sensitivity to image contrast is common to all corner operators that are based on the local autocorrelation of image intensity values and/or image gradient values.

Unlike image intensity gradient values phase congruency values are normalized quantities that have no units associated with them. If the moments are normalized for the number of orientations considered we end up with phase congruency moment values that range between 0 and 1. Being moments these values correspond to phase congruency squared. Accordingly we can use the maximum and minimum phase congruency moments *directly* to establish whether we have a significant edge and/or corner point. It should be emphasized that the minimum and maximum moments provide feature information in their own right; one does not have to look at their ratios. We can define a priori what a significant value of phase congruency moment is, and this value is independent of image contrast.

4 Results

The performance of the phase congruency operator was compared to the Harris operator on a synthetic test image, and on a real scene containing strong

shadows. The results are shown in figures 3 and 4. Raw Harris corner strength and raw phase congruency corner and edge strength images are displayed for comparison. It should be noted that the Harris corner strength values varied by many orders of magnitude across the image depending on contrast. To facilitate the display of the Harris corner strength image what is actually shown here is the *fourth root* of the image. Even after this transformation large variations are still evident. In contrast the phase congruency edge and corner strength images are minimally affected by image contrast and are readily thresholded (in this case with a value of 0.4 on both images) to produce a clear set of features.

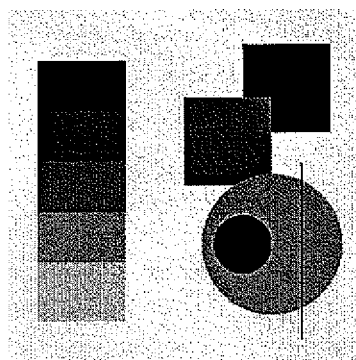
In applying the Harris operator to the synthetic image the standard deviation of the smoothing Gaussian was deliberately set to a large value of 3 pixels to illustrate the problem the operator has in localizing 'T' junctions. Notice how the detected Harris corners are displaced inwards on the left hand grey scale, note also the double response where the line intersects the circle. The phase congruency operator, on the other hand, locates 'T' junctions precisely. Changing the number of filter scales used to compute phase congruency does not affect the localization, it only affects the relative significance of features at different scales. For the real scene the standard deviation of the smoothing Gaussian for the Harris operator was 1 pixel. For both images phase congruency was computed using Gabor filters over 4 scales (wavelengths of 4, 8, 16 and 32 pixels) and over 6 orientations.

Another important point to note is that the phase congruency edge map *includes* the corner map, this is unusual for an edge detector! The fact that the phase congruency corner map is a strict subset of the phase congruency edge map greatly simplifies the integration of data computed from edge and corner information. This facilitates the process of building a model of the scene from point and edge data matched over two or more views. In contrast, if the edge and corner information is computed via separate means, say the Canny and Harris operators respectively, the edge data is unlikely to include the corner data. The Gaussian smoothing that is applied to reduce the influence of noise results in the Canny edges being weakened in strength, and rounded at corner locations.

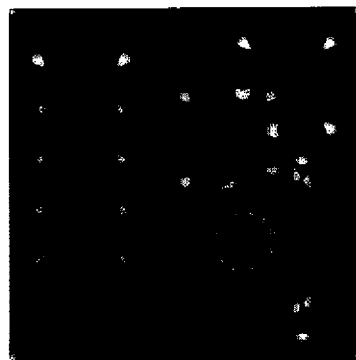
MATLAB code is available for those wishing to replicate the results presented here [21].

5 Conclusion

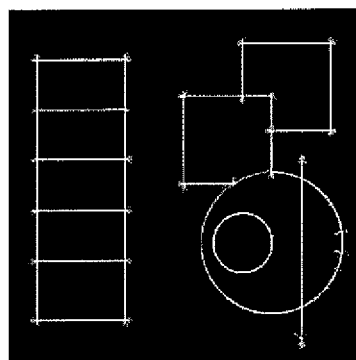
Phase congruency provides a contrast invariant way of identifying features within images. By combining phase congruency information over multiple orientations into a covariance matrix, and calculating the minimum and maximum moments we produce a highly localized operator that can be used to identify both edges and corners in a contrast invariant way. The contrast invariance facilitates the tracking of features over extended image sequences under varying lighting conditions. An additional advantage of the operator is that the phase congruency corner map is a strict *subset* of the phase congruency edge map. This simplifies the integration of data computed from edge and corner information.



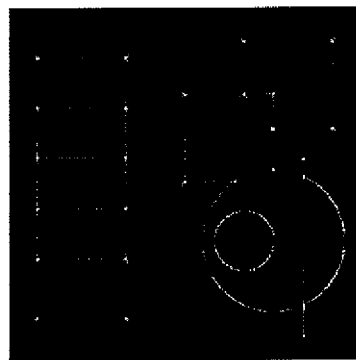
Original image.



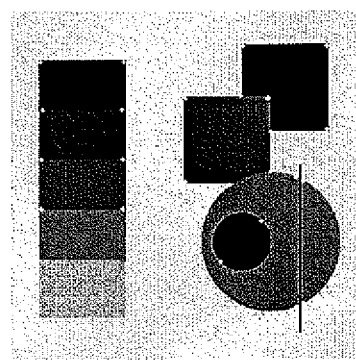
Fourth root of Harris corner strength image ($\sigma = 3$).



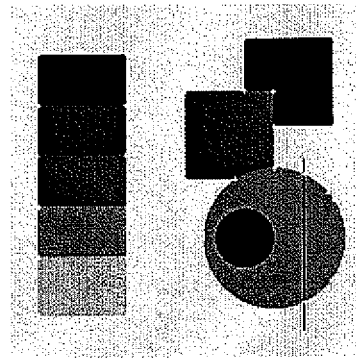
Phase congruency edge strength image.



Phase congruency corner strength image.

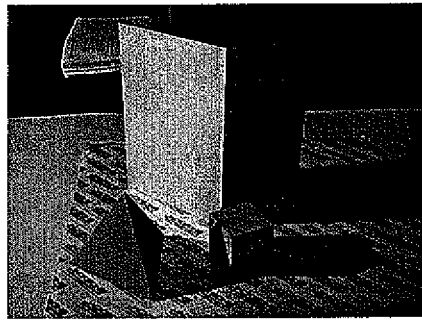


Harris corners with threshold 10^7 (maximum corner strength was 3.4×10^9).

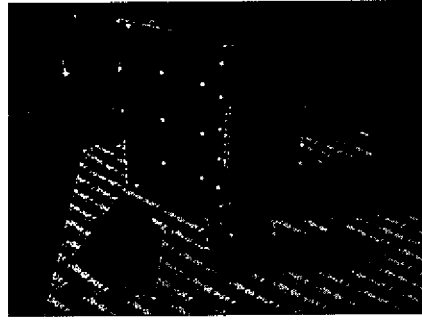


Phase congruency corners with threshold 0.4 (maximum possible phase congruency value is 1).

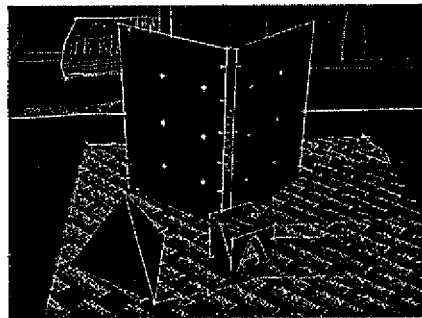
Fig. 3. Comparison of the Harris and phase congruency operators on a test image.



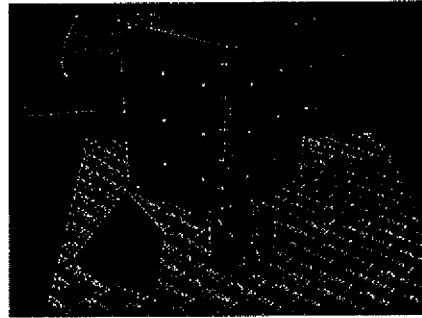
Original image.



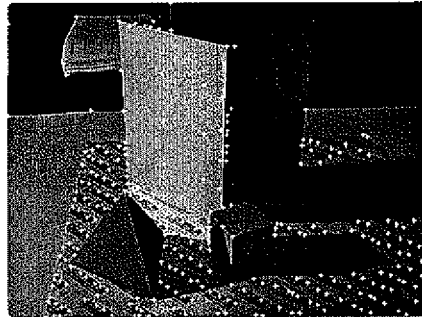
Fourth root of Harris corner strength image ($\sigma = 1$).



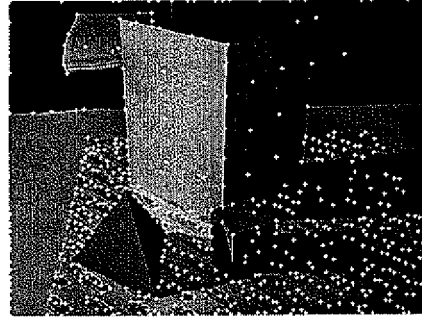
Phase congruency edge strength image.



Phase congruency corner strength image.



Harris corners with threshold 10^8 (maximum corner strength was 1.25×10^{10}).



Phase congruency corners with threshold 0.4 (maximum possible phase congruency value is 1).

Fig. 4. Comparison of the Harris and phase congruency operators on an image with strong shadows.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
2. Moravec, H.P.: Robot rover visual navigation. UMI Research Press (1981)
3. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings, 4th Alvey Vision Conference. (1988) 147–151 Manchester.
4. Beaudet, P.R.: Rotationally invariant image operators. In: International Joint Conference on Artificial Intelligence. (1987) 579–583
5. Kitchen, L., Rosenfeld, A.: Grey-level corner detection. Pattern Recognition Letters (1982) 95–102
6. Cooper, J., Venkatesh, S., Kitchen, L.: Early jump-out corner detectors. PAMI **15** (1993) 823–828
7. Rosenthaler, L., Heitger, F., Kubler, O., Heydt, R.v.: Detection of general edges and keypoints. In: ECCV92, Springer-Verlag Lecture Notes in Computer Science. Volume 588., Springer-Verlag (1992) 78–86 Santa Margherita Ligure, Italy.
8. Robbins, B., Owens, R.: 2D feature detection via local energy. Image and Vision Computing **15** (1997) 353–368
9. Smith, S.M., Brady, J.M.: SUSAN - a new approach to low level image processing. International Journal of Computer Vision **23** (1997) 45–78
10. Morrone, M.C., Ross, J.R., Burr, D.C., Owens, R.A.: Mach bands are phase dependent. Nature **324** (1986) 250–253
11. Kovesi, P.D.: Edges are not just steps. In: Proceedings of the Fifth Asian Conference on Computer Vision. (2002) 822–827 Melbourne.
12. Morrone, M.C., Owens, R.A.: Feature detection from local energy. Pattern Recognition Letters **6** (1987) 303–313
13. Morrone, M.C., Burr, D.C.: Feature detection in human vision: A phase-dependent energy model. Proc. R. Soc. Lond. B **235** (1988) 221–245
14. Owens, R.A., Venkatesh, S., Ross, J.: Edge detection is a projection. Pattern Recognition Letters **9** (1989) 223–244
15. Venkatesh, S., Owens, R.: On the classification of image features. Pattern Recognition Letters **11** (1990) 339–349
16. Kovesi, P.D.: A dimensionless measure of edge significance. In: The Australian Pattern Recognition Society, Conference on Digital Image Computing: Techniques and Applications. (1991) 281–288 Melbourne.
17. Kovesi, P.D.: A dimensionless measure of edge significance from phase congruency calculated via wavelets. In: First New Zealand Conference on Image and Vision Computing. (1993) 87–94 Auckland.
18. Kovesi, P.D.: Invariant Measures of Image Features From Phase Information. PhD thesis, The University of Western Australia (1996)
19. Kovesi, P.D.: Image features from phase congruency. Videre: Journal of Computer Vision Research **1** (1999) 1–26 <http://mitpress.mit.edu/e-journals/Videre/>.
20. Kovesi, P.D.: Phase congruency: A low-level image invariant. Psychological Research **64** (2000) 136–148
21. Kovesi, P.D.: MATLAB functions for computer vision and image analysis (1996–2003) <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/>.
22. Horn, B.K.P.: Robot Vision. McGraw-Hill (1986) New York.

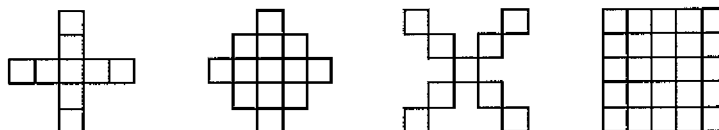
Corners and Junctions Detection



Principal investigator: Robert Laganieri

Morphological Corner Detection

This operator uses a variant of the morphological closing operator, which we have called asymmetrical closing. It consists of the successive application of different morphological transformations using different structuring elements. Each of these structuring elements used to probe the image under study is tuned to affect corners of different orientation and brightness.



The operator can be easily implemented with Intel OpenCV as follows:

```
cvDilate(src, tmp1, cross);
cvErode(tmp1, tmp2, lozeng);
cvDilate(src, dst, ex);
cvErode(dst, tmp1, square);
cvAbsDiff(tmp2, tmp1, dst);
if (threshold > 0)
    cvThreshold(dst, dst, threshold, 255, CV_THRESH_BINARY);
```

Publications:

Robert Laganieri,
Morphological Corner Detection ,
in Proc. International Conference in Computer Vision, pp. 280-285, Bombay, India, January 1998.
PDF [0.223 mb]

Robert Laganieri,
A Morphological Operator for Corner Detection ,
in Pattern Recognition, pp. 1643-1652, November 1998.
Online version

A modified version of this operator has been proposed in:

Shih, Frank Y.; Chuang, Chao-Fa; Gaddipati, Vijayalakshmi,
A modified regulated morphological corner detector ,
in Pattern Recognition Letters, pp. 931-937, April 2005.
Online version

Source code

Available [here](#). The file `comorph.h` contains the operator's actual implementation that can be reused in other applications.

JUDOCA: Junction Detection Operator based on Circumferential Ancors

Participant: Rimón Elias

This junction detection operator defines junctions as points where linear ridges in the gradient domain intersect. The radial lines that compose the junction are therefore identified by searching, in a circular neighborhood, for directional maxima of the intensity gradient. The proposed algorithm operates on two binary edge maps, the computational complexity of the detection process is then considerably reduced.

Algorithm Outline:

1. Apply vertical and horizontal Gaussian derivative filters on image I .
2. Compute gradient magnitude and create two binary images from it:
 1. The first one, B , is created from the imposition of a threshold, t_B , on the gradient image.
 2. The second one, B^+ , contains the points of B that are local maxima in the direction of the gradient.
3. For each point p in B , consider a circle of radius rho centered on this point and obtain the list of candidate points q_i in B^+ that lie on the circumference of this circle (see Figure 1). These so-called circumferential anchor (CA) points are the extremities of potential radial lines for the putative junction.

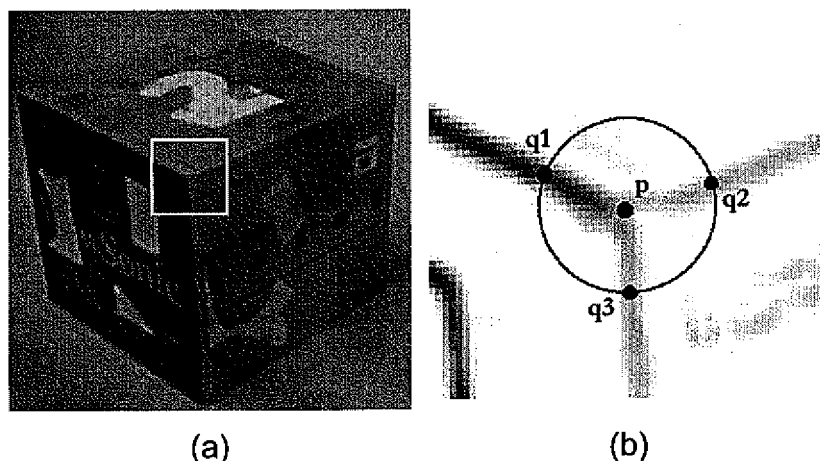


Figure 1: (a) One corner of a box that produces a Y-junction. (b) The junction at p with three circumferential anchors q_1 , q_2 and q_3 (superimposed on the gradient image).

4. For each CA point q_i in the list, consider the set of all points located at a distance less than one pixel to the segment that joins the current CA point to the central point p . This set is used to determine if the corresponding putative junction radial line will be accepted (considering the points in B) and if yes, what the strength of this junction line would be (considering the points in B^+), that is:
 1. To be accepted as a junction line, a continuous path of B points joining the CA point and the central point p must exist. If not, then reject this radial line and repeat the scanning operation with the next CA point.
 2. The strength, $S_j(<p, q_i>)$, of this junction radial line is defined as the sum of the squared distances from the B^+ points in the currently considered set to the $<p, q_i>$ line segment. This strength is normalized by the length of this segment.
5. If the strength of this junction radial line is smaller than a predetermined threshold, s_j , then reject this radial line.

Otherwise, $\langle p, q_i \rangle$ becomes one of the branch of the putative junction at p .

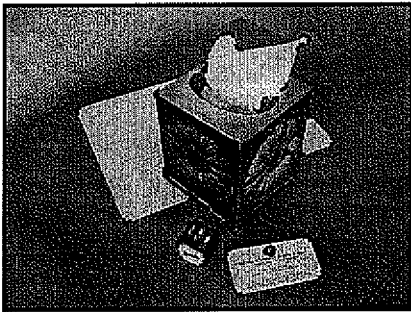
6. If the number of branches found at p is less than 2, then there is no junction at this location. Otherwise, record the orientation of the accepted junction radial lines and set the strength, $S_j(p)$, of this junction as being the minimum across all radial line strengths, i.e.:

$$S_j(p) = \min_i S_j(\langle p, q_i \rangle)$$

Examples:



(Click on image for better resolution)



(Click on image for better resolution)

Publications:

Robert Laganière, Rimon Elias

The Detection of Junction Features in Images ,

in Proc. International Conference on Acoustic, Speech and Signal processing, pp. 573-576, Montreal, Canada, May 2004.

PDF [0.223 mb]

Source code

Available [here](#).

Copyright © 2004 VIVA Lab

[[VIGRA Homepage](#) | [Class Index](#) | [Function Index](#) | [File Index](#) | [Main Page](#)]



Corner Detection



Functions

template<...> void **cornerResponseFunction** (SrcIterator sul, SrcIterator slr, SrcAccessor as, DestIterator dul, DestAccessor ad, double scale)
Find corners in an image (1).

template<...> void **foerstnerCornerDetector** (SrcIterator sul, SrcIterator slr, SrcAccessor as, DestIterator dul, DestAccessor ad, double scale)
Find corners in an image (2).

template<...> void **rohrCornerDetector** (SrcIterator sul, SrcIterator slr, SrcAccessor as, DestIterator dul, DestAccessor ad, double scale)
Find corners in an image (3).

template<...> void **beaudetCornerDetector** (SrcIterator sul, SrcIterator slr, SrcAccessor as, DestIterator dul, DestAccessor ad, double scale)
Find corners in an image (4).

Detailed Description

Measure the 'cornerness' at each pixel. Note: The Kitchen-Rosenfeld detector is not implemented because of its inferior performance. The SUSAN detector is missing because it's patented.

Function Documentation

void beaudetCornerDetector (...)

Find corners in an image (4).

This algorithm implements a corner detector according to [P.R. Beaudet: "*Rotationally Invariant Image Operators*", Proc. Intl. Joint Conf. on Pattern Recognition, Kyoto, Japan, 1978, pp. 579-583].

The algorithm calculates the corner strength as the negative determinant of the **Hessian Matrix**. The local maxima of the corner strength denote the corners in the gray level image.

The source value type must be a linear algebra, i.e. addition, subtraction, and multiplication with itself,

multiplication with doubles and NumericTraits must be defined.

Declarations:

pass arguments explicitly:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    void
    beaudetCornerDetector(SrcIterator sul, SrcIterator slr, SrcAccessor as,
                        DestIterator dul, DestAccessor ad,
                        double scale)
}
```

use argument objects in conjunction with Argument Object Factories:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    inline
    void beaudetCornerDetector(
        triple<SrcIterator, SrcIterator, SrcAccessor> src,
        pair<DestIterator, DestAccessor> dest,
        double scale)
}
```

Usage:

#include "vigra/cornerdetection.hxx"

Namespace: vigra

```
vigra::BImage src(w,h), corners(w,h);
vigra::FImage beaudet_corner_strength(w,h);

// empty corner image
corners.init(0.0);
...

// find corner response at scale 1.0
vigra::beaudetCornerDetector(srcImageRange(src), destImage(beaudet_corner_strength),
                            1.0);

// find local maxima of corner response, mark with 1
vigra::localMaxima(srcImageRange(beaudet_corner_strength), destImage(corners));
```

Required Interface:

```
SrcImageIterator src_upperleft, src_lowerright;
DestImageIterator dest_upperleft;

SrcAccessor src_accessor;
DestAccessor dest_accessor;

SrcAccessor::value_type u = src_accessor(src_upperleft);
```

```
double d;

u = u + u
u = u - u
u = u * u
u = d * u

dest_accessor.set(u, dest_upperleft);
```

void cornerResponseFunction (...)

Find corners in an image (1).

This algorithm implements the so called 'corner response function' to measure the 'cornerness' of each pixel in the image, according to [C.G. Harris and M.J. Stevens: "A Combined Corner and Edge Detector", Proc. of 4th Alvey Vision Conference, 1988]. Several studies have found this to be a very robust corner detector, although it moves the corners somewhat into one region, depending on the scale.

The algorithm first determines the structure tensor at each pixel by calling `structureTensor()`. Then the entries of the structure tensor are combined as

$$\text{CornerResponse} = \det(\text{StructureTensor}) - 0.04 \text{tr}(\text{StructureTensor})^2 = AB - C^2 - 0.04(A+B)^2$$

The local maxima of the corner response denote the corners in the gray level image.

The source value type must be a linear algebra, i.e. addition, subtraction, and multiplication with itself, multiplication with doubles and NumericTraits must be defined.

Declarations:

pass arguments explicitly:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    void
    cornerResponseFunction(SrcIterator sul, SrcIterator slr, SrcAccessor as,
                          DestIterator dul, DestAccessor ad,
                          double scale)
}
```

use argument objects in conjunction with Argument Object Factories:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    inline
    void cornerResponseFunction(
        triple<SrcIterator, SrcIterator, SrcAccessor> src,
        pair<DestIterator, DestAccessor> dest,
```

```

        double scale)
    }

```

Usage:

#include "vignra/cornerdetection.hxx"

Namespace: vignra

```

vignra::BImage src(w,h), corners(w,h);
vignra::FImage corner_response(w,h);

// empty corner image
corners.init(0.0);
...

// find corner response at scale 1.0
vignra::cornerResponseFunction(srcImageRange(src), destImage(corner_response),
                               1.0);

// find local maxima of corner response, mark with 1
vignra::localMaxima(srcImageRange(corner_response), destImage(corners));

// threshold corner response to keep only strong corners (above 400.0)
transformImage(srcImageRange(corner_response), destImage(corner_response),
               vignra::Threshold<double, double>{
                   400.0, std::numeric_limits<double>::max(), 0.0, 1.0});

// combine thresholding and local maxima
vignra::combineTwoImages(srcImageRange(corners), srcImage(corner_response),
                         destImage(corners), std::multiplies<float>());

```

Required Interface:

```

SrcImageIterator src_upperleft, src_lowerright;
DestImageIterator dest_upperleft;

SrcAccessor src_accessor;
DestAccessor dest_accessor;

SrcAccessor::value_type u = src_accessor(src_upperleft);
double d;

u = u + u
u = u - u
u = u * u
u = d * u

dest_accessor.set(u, dest_upperleft);

```

void foerstnerCornerDetector (...)

Find corners in an image (2).

This algorithm implements the so called 'Foerstner Corner Detector' to measure the 'cornerness' of eac

pixel in the image, according to [W. Förstner: *"A feature based correspondence algorithms for image matching"*, Intl. Arch. Photogrammetry and Remote Sensing, vol. 24, pp 160-166, 1986]. It is also known as the "Plessey Detector" by Harris. However, it should not be confused with the "**Corner Response Function**", another detector invented by Harris.

The algorithm first determines the structure tensor at each pixel by calling `structureTensor()`. Then the entries of the structure tensor are combined as

$$\text{FoerstnerCornerStrength} = \frac{\det(\text{StructureTensor})}{\text{tr}(\text{StructureTensor})} = \frac{AB - C^2}{A + B}$$

The local maxima of the corner strength denote the corners in the gray level image. Its performance is similar to the `cornerResponseFunction()`.

The source value type must be a division algebra, i.e. addition, subtraction, multiplication, and division with itself, multiplication with doubles and `NumericTraits` must be defined.

Declarations:

pass arguments explicitly:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    void
    foerstnerCornerDetector(SrcIterator sul, SrcIterator slr, SrcAccessor as,
                          DestIterator dul, DestAccessor ad,
                          double scale)
}
```

use argument objects in conjunction with Argument Object Factories:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    inline
    void foerstnerCornerDetector(
        triple<SrcIterator, SrcIterator, SrcAccessor> src,
        pair<DestIterator, DestAccessor> dest,
        double scale)
}
```

Usage:

#include "vigra/cornerdetection.hxx"

Namespace: `vigra`

```
vigra::BImage src(w,h), corners(w,h);
vigra::FImage foerstner_corner_strength(w,h);

// empty corner image
corners.init(0.0);
...
```

```
// find corner response at scale 1.0
vigra::foerstnerCornerDetector(srcImageRange(src), destImage(foerstner_corner_strength)
    1.0);

// find local maxima of corner response, mark with 1
vigra::localMaxima(srcImageRange(foerstner_corner_strength), destImage(corners));
```

Required Interface:

```
SrcImageIterator src_upperleft, src_lowerright;
DestImageIterator dest_upperleft;

SrcAccessor src_accessor;
DestAccessor dest_accessor;

SrcAccessor::value_type u = src_accessor(src_upperleft);
double d;

u = u + u
u = u - u
u = u * u
u = u / u
u = d * u

dest_accessor.set(u, dest_upperleft);
```

void rohrCornerDetector (...)

Find corners in an image (3).

This algorithm implements yet another structure tensor-based corner detector, according to [K. Rohr: *"Untersuchung von grauwertabhängigen Transformationen zur Ermittlung der optischen Flusses in Bildfolgen"*, Diploma thesis, Inst. für Nachrichtensysteme, Univ. Karlsruhe, 1987, see also K. Rohr: *"Modelling and Identification of Characteristic Intensity Variations"*, Image and Vision Computing 10:2 (1992) 66-76 and K. Rohr: *"Localization Properties of Direct Corner Detectors"*, J. of Mathematical Imaging and Vision 4:2 (1994) 139-150].

The algorithm first determines the structure tensor at each pixel by calling **structureTensor()**. Then the entries of the structure tensor are combined as

$$\text{RohrCornerStrength} = \det(\text{StructureTensor}) = AB - C^2$$

The local maxima of the corner strength denote the corners in the gray level image. Its performance is similar to the **cornerResponseFunction()**.

The source value type must be a linear algebra, i.e. addition, subtraction, and multiplication with itself, multiplication with doubles and NumericTraits must be defined.

Declarations:

pass arguments explicitly:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    void
    rohrCornerDetector(SrcIterator sul, SrcIterator slr, SrcAccessor as,
                      DestIterator dul, DestAccessor ad,
                      double scale)
}
```

use argument objects in conjunction with Argument Object Factories:

```
namespace vigra {
    template <class SrcIterator, class SrcAccessor,
              class DestIterator, class DestAccessor>
    inline
    void rohrCornerDetector(
        triple<SrcIterator, SrcIterator, SrcAccessor> src,
        pair<DestIterator, DestAccessor> dest,
        double scale)
}
```

Usage:

#include "vigra/cornerdetection.hxx"

Namespace: vigra

```
vigra::BImage src(w,h), corners(w,h);
vigra::FImage rohr_corner_strength(w,h);

// empty corner image
corners.init(0.0);
...

// find corner response at scale 1.0
vigra::rohrCornerDetector(srcImageRange(src), destImage(rohr_corner_strength),
                          1.0);

// find local maxima of corner response, mark with 1
vigra::localMaxima(srcImageRange(rohr_corner_strength), destImage(corners));
```

Required Interface:

```
SrcImageIterator src_upperleft, src_lowerright;
DestImageIterator dest_upperleft;

SrcAccessor src_accessor;
DestAccessor dest_accessor;

SrcAccessor::value_type u = src_accessor(src_upperleft);
double d;

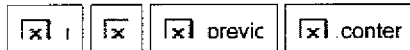
u = u + u
u = u - u
u = u * u
```

```
u = d * u
```

```
dest_accessor.set(u, dest_upperleft);
```

© Ullrich Köthe (koethe@informatik.uni-hamburg.de)
Cognitive Systems Group, University of Hamburg,
Germany

html generated using doxygen and
Python
VIGRA 1.5.0 (7 Dec 2006)



Next: [Line segment detection](#) **Up:** [Feature detection](#) **Previous:** [Displaying a corner map](#) [Contents](#)

The Harris corner detector

```
#include <gandalf/vision/harris_corner.h>
```

The Harris corner detector [#!Harris:Stephens:ALVEY88!#] computes the locally averaged moment matrix computed from the image gradients, and then combines the eigenvalues of the moment matrix to compute a corner ``strength'', of which maximum values indicate the corner positions. Here is an example code fragment using the Harris corner detector.

```
Gan_Image *pImage; /* declare image from which corners will be computed */
Gan_Mask1D *pFilter; /* convolution mask */
Gan_CornerFeatureMap CornerMap; /* declare corner map */

/* ... fill image ... */

/* initialise corner map */
gan_corner_feature_map_form ( &CornerMap,
                             1000 ); /* initial limit on number of corners */

/* create convolution mask */
pFilter = gan_gauss_mask_new ( GAN_FLOAT, 1.0, 9, 1.0, NULL );

/* apply Harris corner detector */
gan_harris_corner_q ( pImage, /* input image */
                     NULL, /* or binary mask of pixels to be processed */
                     NULL, NULL, /* or image pre-smoothing masks */
                     pFilter, pFilter, /* gradient smoothing */
                     0.04, /* kappa used in computing corner strength */
                     0.04, /* corner strength threshold */
                     NULL, /* or affine coordinate transformation */
                     0, /* status value to assign to each corner */
                     NULL, /* or pointer to camera structure defining
                             distortion model */
                     NULL, /* or parameters of local feature map */
                     &CornerMap ); /* result corner map */

/* free convolution mask and corner map */
gan_mask1D_free ( pFilter );
gan_corner_feature_map_free ( &CornerMap );
```

2006-03-17

Robust point-based feature fingerprint segmentation algorithm

Chaohong Wu, Sergey Tulyakov and Venu Govindaraju

Center for Unified Biometrics and Sensors (CUBS)
SUNY at Buffalo, USA

Abstract. A critical step in automatic fingerprint recognition is the accurate segmentation of fingerprint images. The objective of fingerprint segmentation is to decide which part of the images belongs to the foreground containing features for recognition and identification, and which part to the background with the noisy area around the boundary of the image. Unsupervised algorithms extract blockwise features. Supervised method usually first extracts point features like coherence, average gray level, variance and Gabor response, then a Fisher linear classifier is chosen for classification. This method provides accurate results, but its computational complexity is higher than most of unsupervised methods. This paper proposes using Harris corner point features to discriminate foreground and background. Shifting a window in any direction around the corner should give a large change in intensity. We observed that the strength of Harris point in the foreground area is much higher than that of Harris point in background area. The underlying mechanism for this segmentation method is that boundary ridge endings are inherently stronger Harris corner points. Some Harris points in noisy blobs might have higher strength, but it can be filtered as outliers using corresponding Gabor response. The experimental results proved the efficiency and accuracy of new method are markedly higher than those of previously described methods.

1 Introduction

The accurate segmentation of fingerprint images is key component to achieve high performance in automatic fingerprint recognition systems. If more background areas are included into segmented fingerprint of interest, more false features are possibly introduced into detected feature set; If some parts of foreground are excluded, useful feature points may be missed. There are two types of fingerprint segmentation algorithms: unsupervised and supervised. Unsupervised algorithms extract blockwise features such as local histogram of ridge orientation [1, 2], gray-level variance, magnitude of the gradient in each image block [3], Gabor feature [4, 5]. Practically, the presence of noise, low contrast area, and inconsistent contact of a fingertip with the sensor may result in loss of minutiae or more spurious minutiae. Supervised method usually first extracts several features like coherence, average gray level, variance and Gabor response [5–7], then a simple linear classifier is chosen for classification. This method provides accurate results, but its computational complexity is higher than most unsupervised methods.

Segmentation in low quality images faces several challenging technical problems. First problem is the presence of noise caused by dust and grease on the surface of live-scan fingerprint scanners. Second problem is ghost images of fingerprints remaining

from the previous image acquisition [7]. Third problem is low contrast fingerprint ridges generated through inconsistent contact press or dry/wet finger surface. Fourth problem is indistinct boundary if the features in the fixed size of window are used. Final problem is segmentation features being sensitive to the quality of image.

This paper proposes using Harris corner point features [8, 9] to discriminate foreground and background. The Harris corner detector was developed originally as features for motion tracking, it can reduce significantly amount of computation compared to tracking every pixel. It is translation and rotation invariant but not scale invariant. We found that the strength of a Harris point in the foreground area is much higher than that of a Harris point in the background area. Some Harris points in noisy blobs might have higher strength, but it can be filtered as outliers using corresponding Gabor response. The experimental results proved the efficiency and accuracy of new method are much better than those of previously described methods. Furthermore, this segmentation algorithm can detect accurate boundary of fingerprint ridge regions, which is very useful in removing spurious boundary minutiae, and most current segmentation methods can not provide consistent boundary minutiae filtering.

2 Features for Fingerprint Segmentation

Feature selection is the first step for designing fingerprint segmentation algorithm. There are two general types of features used for fingerprint segmentation, i.e., block features and pointwise features. In [5, 10] selected point features include local mean, local variance or standard deviation, and Gabor response of the fingerprint image. Local mean is calculated as $Mean = \sum_w I$, local variance is calculated as $Var = \sum_w (I - Mean)^2$, w is window size centered the processed pixel. The Gabor response is the smoothed sum of Gabor energies for eight Gabor filter responses. Usually the Gabor response is higher in the foreground region than that in the background region. The coherence feature indicates how strong the local window gradients centered the processed point in the same dominant orientation. Usually the coherence will be much higher in the foreground than in the background, but it may be influenced significantly by boundary signal and noisy signal. Therefore, single coherence feature is not sufficient for robust segmentation. Systematic combination of those features is necessary.

$$Coh = \frac{|\sum_w (G_{s,x}, G_{s,y})|}{|\sum_w (G_{s,x}, G_{s,y})|} = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \quad (1)$$

Because pointwise-based segmentation method is time consuming, blockwise features are usually used in the commercial automatic fingerprint recognition systems. Block mean, block standard deviation, block gradient histogram [1, 2], block average magnitude of the gradient [11] are most common block features for fingerprint segmentation. In [12] gray-level pixel intensity-derived feature called block clusters degree(ChuD) is introduced. *ChuD* measures how well the ridge pixels are clustering.

$$E(x, y) = \log \left\{ \int_r \int_\theta |F(r, \theta)|^2 \right\} \quad (2)$$

Texture features, such as Fourier spectrum energy [6], Gabor features [4, 13] and Gaussian-Hermite Moments [14], have been applied to fingerprint segmentation. Ridges and valleys in a fingerprint image are generally observed to possess a sinusoidal-shaped plane wave with a well-defined frequency and orientation [15], and non-ridge regions does not hold this surface wave model. In the areas of background and noisy regions, it is assumed that there is very little structure and hence very little energy content in the Fourier spectrum. Each value of energy image $E(x,y)$ indicates the energy content of the corresponding block. The fingerprint region may be differentiated from the background by thresholding the energy image. The logarithm values of the energy is used to convert the large dynamic range to a linear scale(Equation 2). The region mask is obtained by thresholding $E(x,y)$. However, uncleaned trace finger ridges and straight stripes are unfortunately included into regions of interest (Figure 1(c)).

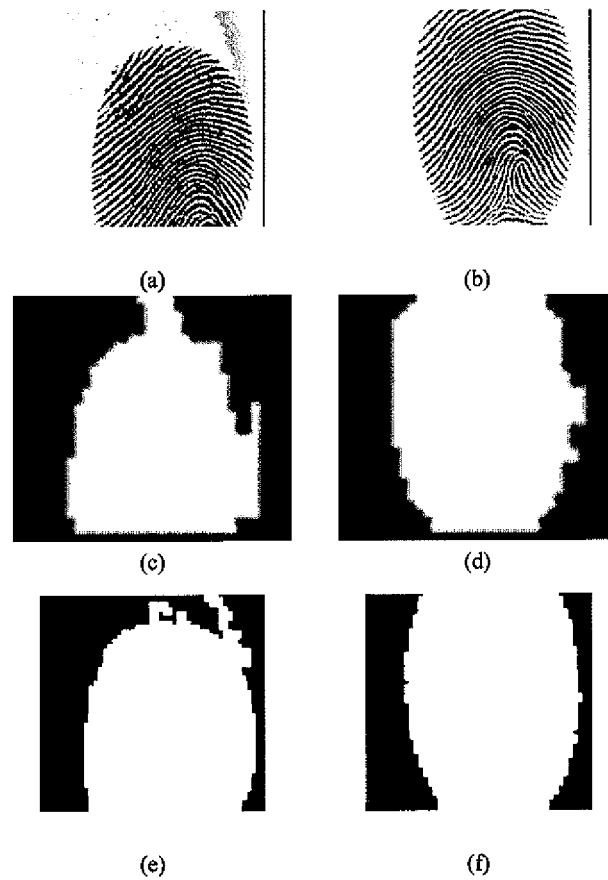


Fig. 1. (a) and (b) are two original images, (c) and (d) are FFT energy maps for images (a) and (b), (e) and (f) are Gabor energy maps for images (a) and (b), respectively

Gabor filter-based segmentation algorithm is now most often used method [4, 13]. An even symmetric Gabor filter has the following spatial form:

$$g(x, y, \theta, f, \sigma_x, \sigma_y) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_\theta) \quad (3)$$

For each block of size $W \times W$ centered at (x, y) , 8 directional Gabor features are computed for each block, the standard deviation of 8 Gabor features is utilized for segmentation. The formula for calculating the magnitude of Gabor feature is defined as,

$$G(X, Y, \theta, f, \sigma_x, \sigma_y) = \left| \sum_{x_0=-w/2}^{(w/2)-1} \sum_{y_0=-w/2}^{(w/2)-1} I(X + x_0, Y + y_0) g(x_0, y_0, \theta, f, \sigma_x, \sigma_y) \right| \quad (4)$$

However, fingerprint images with low contrast or false traces ridges or noisy complex background can not be segmented correctly by Gabor filter-based method (Figure 1(e)).

In [14], similarity is found between Hermite moments and Gabor filter. Gaussian-Hermite Moments has been successfully used to segment fingerprint images in [14]. Orthogonal moments use orthogonal polynomials as transform kernels and produce minimal information redundancy, Gaussian Hermite moments(GHM) can represent local texture feature without minimal noise effect.

3 Harris Corner Points

3.1 Review of Harris-corner-points

We propose using Harris corner point features [8, 9] to discriminate foreground and background. The Harris corner detector was developed originally as features for motion tracking, it can reduce significantly amount of computation compared to tracking every pixel. Shifting a window in any direction around the corner should give a large change in intensity. Corner points provide repeatable points for matching, so some efficient methods have been designed [8, 9]. Gradient is ill defined at a corner, so edge detectors perform poorly at corners. However, in the region around a corner, gradient has two or more different values. The corner point can be easily recognized by looking a small window. Shifting a window around a corner point in any direction should give a large change in gray-level intensity,

Given a point $I(x, y)$, and a shift $(\Delta x, \Delta y)$, the auto-correlation function E is defined as:

$$E(x, y) = \sum_{w(x, y)} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \quad (5)$$

where $w(x, y)$ is window function centered on image point (x, y) . For a small shifts $[\Delta x, \Delta y]$, the shifted image is approximated by a *Taylor expansion* truncated to the first order terms,

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (6)$$

where $I_x(x_i, y_i)$ and $I_y(x_i, y_i)$ denote the partial derivatives in x and y, respectively. Substituting approximation Equation 6 into Equation 5 yields,

$$\begin{aligned}
E(x, y) &= \sum_{w(x, y)} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \\
&= \sum_{w(x, y)} \left(I(x_i, y_i) - I(x_i, y_i) - [I_x(x_i, y_i)I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= \sum_{w(x, y)} \left(-[I_x(x_i, y_i)I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= \sum_{w(x, y)} \left([I_x(x_i, y_i)I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= [\Delta x, \Delta y] \begin{bmatrix} \sum_w (I_x(x_i, y_i))^2 & \sum_w I_x(x_i, y_i)I_y(x_i, y_i) \\ \sum_w I_x(x_i, y_i)I_y(x_i, y_i) & \sum_w (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\
&= [\Delta x, \Delta y] M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
\end{aligned} \tag{7}$$

That is,

$$E(\Delta x, \Delta y) = [\Delta x, \Delta y] M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{8}$$

where $M(x, y)$ is a 2×2 matrix computed from image derivatives, called auto-correlation matrix which captures the intensity structure of the local neighborhood.

$$M = \sum_{x, y} w(x, y) \begin{bmatrix} (I_x(x_i, y_i))^2 & I_x(x_i, y_i)I_y(x_i, y_i) \\ I_x(x_i, y_i)I_y(x_i, y_i) & (I_y(x_i, y_i))^2 \end{bmatrix} \tag{9}$$

3.2 Strength of Harris-corner points of a Fingerprint Image

In order to detect interest points, the original measure of corner response in [8] is :

$$R = \frac{\det(M)}{\text{Trace}(M)} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \tag{10}$$

The auto-correlation matrix (M) captures the structure of the local neighborhood. Based on eigenvalues (λ_1, λ_2) of M, interest points are located where there are two strong eigen values and the corner strength is a local maximum in a 3×3 neighborhood. To avoid the explicit eigenvalue decomposition of M, $\text{Trace}(M)$ is calculated as $I_x^2 + I_y^2$, $\text{Det}(m)$ is calculated as $I_x^2 I_y^2 - (I_x I_y)^2$, and

$$R = \text{Det}(m) - k \times \text{Trace}(M)^2 \tag{11}$$

To segment the fingerprint area (foreground) from the background, the following “corner strength” measure is used, because there is one undecided parameter k in equation(11).

$$R = \frac{I_x^2 I_y^2 - I_{xy}^2}{I_x^2 + I_y^2} \tag{12}$$

3.3 Harris-corner-points based Fingerprint Image Segmentation

We found that the strength of a Harris point in the fingerprint area is much higher than that of a Harris point in background area, because boundary ridge endings inherently possess higher corner strength. Most high quality fingerprint images can be easily segmented by choosing appropriate threshold value. In Figure 2, a corner strength of 300 is selected to distinguish corner points in the foreground from those in the background. Convex hull algorithm is used to connect harris corner points located in the foreground boundary.

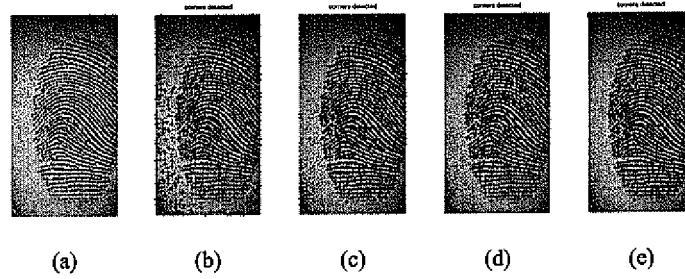


Fig. 2. a fingerprint with harris corner strength of (b)10, (c)60, (d)200, and (e)300. This fingerprint can be successfully segmented using corner response threshold of 300.

It appears relatively easy for us to segment fingerprint images for following image enhancement, feature detection and matching. However, two technical problems need to be solved. First, different “corner strength” thresholds are necessary to achieve good segmentation results for different qualities images based on image characteristic analysis. Second, Some Harris points in noisy blobs might have higher strength, it can not be segmented by choosing simply one threshold. When single threshold is applied to all the fingerprint images in one whole database, not all the corner points in the background in a fingerprint image are removed, some corner points in noisy regions can not be thresholded even using high threshold value (Figure 3). In order to deal with such situations, we implemented a heuristic selection algorithm using corresponding Gabor response (Figure 4).

4 Experimental Results

The proposed methodology is tested on FVC2002 DB1 and DB4, each database consists of 800 fingerprint images (100 distinct fingers, 8 impressions each). Image size is 374×388 and the resolution is 500dpi. To evaluate the methodology of adapting a gaussian kernel to the local ridge curvature of a fingerprint image, we modified Gabor-based fingerprint enhancement algorithm [15, 16] with two kernel sizes: the smaller one in high-curvature regions and the larger one in pseudo-parallel ridge regions, minutiae are



Fig. 3. a fingerprint with harris corner strength of (a)100, (b)500, (c)1000, (d) 1500 and (e)3000. Some noisy corner points can not be filtered completely even using corner response threshold of 3000.

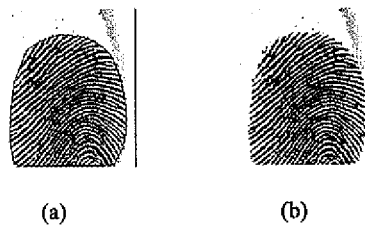


Fig. 4. Segmentation result and final feature detection result for the image shown in the Figure 1(a). (a) Segmented fingerprint marked with boundary line, (b) final detected minutiae.

detected using chaincode-based contour tracing [17], the fingerprint matcher developed by Jea et al. [18] is used for performance evaluation.

Our methodology has been tested on low quality images from FVC2002. To validate the efficiency of proposed segmentation method, current widely-used Gabor filter-based segmentation algorithm [4, 13] and NIST segmentation [19] are utilized for comparison.

The proposed segmentation method have a remarkable advantage over current methods in terms of boundary spurious minutiae filtering. Figure 5 (a) and (b) show unsuccessful boundary minutiae filtering using NIST method [19], which is implemented by removing spurious minutiae pointing to invalid block and removing spurious minutiae near invalid blocks, and invalid blocks are defined as blocks with no detectable ridge flow. However, boundary blocks are more complicated, so the method in [19] fails to remove most boundary minutiae. In Figure 5 (c) and (d) show the filtering results of proposed method. In comparison of Figure 5(a) against (c) and (b) and (d), 30 and 17 boundary minutiae are filtered, respectively. Performance evaluations for FVC2002 DB1 and DB4 are shown in Figure 6. For DB1, ERR for false boundary minutiae filtering using proposed segmented mask is 0.0106 and EER for NIST boundary Filtering is 0.0125. For DB4, ERR for false boundary minutiae filtering using proposed segmented mask is 0.0453 and EER for NIST boundary Filtering is 0.0720.



Fig. 5. Boundary spurious minutiae filtering. (a) and (b) incomplete filtering using NIST method, (c) and (d) proposed boundary filtering.

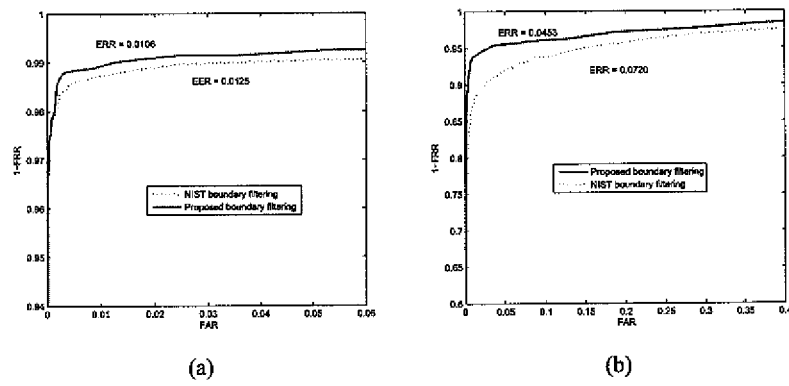


Fig. 6. ROC curves for (a) FVC2002 DB1 and (b) FVC2002 DB4.

5 Conclusions

In this paper, a robust interest point based fingerprint segmentation is proposed for fingerprints of varied image qualities. The experimental results compared with those of previous methods validate that our algorithm has better performance even for low quality images, in terms of including less background and excluding less foreground. In addition, this robust segmentation algorithm is capable of filtering efficiently spurious boundary minutiae.

References

1. Mehtre, B.M., Chatterjee, B.: Segmentation of fingerprint images – a composite method. *Pattern Recognition* **22**(4) (1989) 381–385
2. Mehtre, B.M., Murthy, N.N., Kapoor, S., Chatterjee, B.: Segmentation of fingerprint images using the directional image. *Pattern Recognition* **20**(4) (1987) 429–435
3. Ratha, N.K., Chen, S., Jain, A.K.: Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition* **28**(11) (1995) 1657–1672
4. Alonso-Fernandez, F., Fierrez-Aguilar, J., Ortega-Garcia, J.: An enhanced gabor filter-based segmentation algorithm for fingerprint recognition systems. In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005)*. (2005) 239–244
5. Bazen, A., Gerez, S.: Segmentation of fingerprint images. In: *Proc. Workshop on Circuits Systems and Signal Processing (ProRISC 2001)*. (2001) pp. 276–280
6. Pais Barreto Marques, A.C., Gay Thome, A.C.: A neural network fingerprint segmentation method. In: *Fifth International Conference on Hybrid Intelligent Systems (HIS 05)*. (2005) 6 pp.
7. Zhu, E., Yin, J., Hu, C., Zhang, G.: A systematic method for fingerprint ridge orientation estimation and image segmentation. *Pattern Recognition* **39**(8) (2006) 1452–1472
8. Harris, C., Stephens, M.: A combined corner and edge detector. In: *Proc. in Alvey Vision Conference*. (1988) pp. 147–151
9. Mikolajczyk, K., Schmid, C.: Scale affine invariant interest point detectors. *International Journal of Computer Vision* **60**(1) (2004) pp.63–86
10. Klein, S., Bazen, A.M., Veldhuis, R.: “fingerprint image segmentation based on hidden markov models”. In: *13th Annual workshop in Circuits, Systems and Signal Processing*, in *Proc. ProRISC 2002*. (2002)
11. Maio, D., Maltoni, D.: Direct gray-scale minutiae detection in fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(1) (1997) 27–40
12. Chen, X., Tian, J., Cheng, J., Yang, X.: Segmentation of fingerprint images using linear classifier. *EURASIP Journal on Applied Signal Processing* **2004**(4) (2004) 480–494 doi:10.1155/S1110865704309194.
13. Shen, L., Kot, A., Koo, W.: Quality measures of fingerprint images. In: *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication*. (2001) pp. 266–271
14. Wang, L., Suo, H., Dai, M.: Fingerprint image segmentation based on gaussian-hermite moments. In: *Advanced Data Mining and Applications, LNCS3584*. (2005) 446–454
15. Hong, L., Wan, Y., Jain, A.K.: “Fingerprint image enhancement: Algorithms and performance evaluation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8) (1998) 777–789
16. Wu, C., Govindaraju, V.: Singularity preserving fingerprint image adaptive filtering. In: *International Conference on Image Processing*. (2006) 313–316

17. Wu, C., Shi, Z., Govindaraju, V.: Fingerprint image enhancement method using directional median filter. In: Biometric Technology for Human Identification. Volume 5404., SPIE (2004) 66–75
18. Jea, T.Y., Chavan, V.S., Govindaraju, V., Schneider, J.K.: Security and matching of partial fingerprint recognition systems. In: SPIE Defense and Security Symposium. Volume 5404., SPIE (2004)
19. Watson, C.I., Garris, M.D., Tabassi, E., Wilson, C.L., McCabe, R.M., Janet, S.: “User’s Guide to NIST Fingerprint Image Software2(NFIS2)”. NIST (2004)
20. Otsu, N.: A threshold selection method from gray level histograms. IEEE Transactions on Systems, Man and Cybernetics 9 (1979) 62–66

Structure adaptive anisotropic image filtering

G.Z. Yang^{a,*}, P. Burger^b, D.N. Firmin^a, S.R. Underwood^a

^aMagnetic Resonance Unit, Royal Brompton Hospital, London SW3 6NP, UK

^bDepartment of Computing, Imperial College, London SW7 2AZ, UK

Received 15 December 1994; revised 8 June 1995

Abstract

Noise filtering of images is essentially a smoothing process, and it is an issue that has been addressed for many years. The most commonly used low-pass filtering methods blur important image structures such as edges and lines, and thus reduce image contrast and damage image fidelity. This paper presents a structure adaptive anisotropic filtering technique with its application to processing magnetic resonance images. It differs from other techniques in that, instead of using local gradients as a means of controlling the anisotropism of filters, it uses both a local intensity orientation and an anisotropic measure of level contours to control the shape and extent of the filter kernel. This ensures that corners and junctions are well preserved throughout the filtering process. The following two aspects of the proposed technique demonstrate the advantage of using this filtering method. Firstly, the use of local orientation detection provides a robust and convenient way for shaping the filter kernel. Secondly, the structural adaptiveness of the filtering process ensures that salient image features are non-symmetrically enhanced.

Keywords: Adaptive filtering; Orientation detection; Magnetic resonance imaging

1. Introduction

In medical imaging one is often faced with the problem of compromising between image quality and data acquisition time. To achieve a good Signal-to-Noise Ratio (SNR), time domain averaging is usually used. This prolongs the imaging time, and the method itself is not suited for studies that rely on temporal resolution to reveal important anatomical and functional information. While providing an ultimate solution to the problem mentioned above, improvement of acquisition hardware is often restrained by the practicality and costs involved. In this situation, post processing of image data is needed to improve the clarity of object details. Over the years, two broad categories of technique, linear and non-linear filtering methods, have been used for this purpose. The linear approach has the advantage of simplicity, but it does not respect regional structure and the resulting images often appear blurry and diffused. This undesirable effect can be reduced to a certain extent by the use of the non-linear scheme in which local structures and statistics are taken into account during the filtering process [1–3]. Because of their adaptiveness to local

image features, they are referred to as the *adaptive methods*.

The idea of adaptive smoothing itself is not new, and many different approaches have been proposed over the years. A detailed overview and evaluation of some of the earlier methods can be found in [4]. Lev, Zucker and Rosenfeld [5] performed one of the earliest studies in this field. They suggested applying at each image point a weighted mask with coefficients that are based on the differences between the value at the centre point and those of its neighbours. Similar approaches can also be found elsewhere [2,6]. The major drawback of these iterative smoothing methods is that their convergence properties are difficult to establish. A more recent approach, proposed by Blake and Zisserman [7], used weak continuity constraints to allow discontinuities in a piecewise continuous reconstruction of a noisy signal. The convergence behaviour was well studied and results provided were promising, but unfortunately it was accomplished with a formidable computational cost. By casting the problem in terms of a heat equation in an anisotropic medium, Perona and Malik [8] presented an anisotropic diffusion scheme. The principal aim of their method is to enhance edges and facilitate region segmentation, however, when small iteration steps are

* Corresponding author. Email: gzy@doc.ic.ac.uk

used it can also be considered as a useful tool for image noise filtering [9].

When developing a filtering method for image data, the following two basic requirements should generally be observed: *efficiency* and *fidelity*. Efficiency requires noise to be effectively removed and the morphological definition of the image to be enhanced, whereas fidelity demands detailed image structures to be preserved and no artifacts are generated during the filtering process. Based on these two criteria, in this paper we introduce a structure adaptive anisotropic filtering scheme with its application to processing magnetic resonance images. It differs from other techniques in that, instead of using local gradients as a means of controlling the anisotropy of the filters, it uses the local intensity orientation of level contours and an anisotropic measure to control the shape and the extent of the filter kernel. The relationship between the proposed technique and other methods will be discussed, and a qualitative evaluation of the algorithm will be performed by using a synthetic image with various degrees of additive noise.

2. Basic principle of the algorithm

The most commonly used filtering algorithm computes a new image $f_s(x)$ by applying, at each point x_0 , a kernel $k(x_0, x)$ to the original image $f(x)$ so that:

$$f_s(x_0) = \frac{1}{\mu(x_0)} \int \dots \int_{\Omega} k(x_0, x) f(x) dx \quad (1)$$

where:

$$\mu(x_0) = \int \dots \int_{\Omega} k(x_0, x) dx \quad (2)$$

is a normalization factor. In Eq. (1) $x = (x_1, x_2, \dots, x_m)$ is an m dimensional position vector. It is worth mentioning at this stage that the basic principle of the algorithm proposed is applicable to m dimensional pictures, but applications provided in this paper will be restricted to 2D and 3D magnetic resonance images. When function k in Eq. (1) is invariant over space, it uniformly smooths out image noise without paying any attention to image details. The basic idea behind the proposed algorithm is that the kernel $k(x_0, x)$ used in Eq. (1) should be made variable and allow to be shaped or scaled according to local features within the neighbourhood Ω of x_0 . If one takes local gradients as the primary concern, then intuitively one requires the kernel to be broad where the gradient is weak so that uniform regions can be smoothed out. On the other hand, at places where the gradient is strong the kernel should be narrow so that edges and corners are preserved.

When a Gaussian kernel is adopted, the relationship stated above can be expressed analytically by the

following equation:

$$k(x_0, x) = \rho(x - x_0) e^{-|x - x_0|^2 / \sigma^2(|\nabla f(x_0)|)} \quad (3)$$

where $\rho(x)$ is a positive and rotationally symmetric cut-off function that satisfies the condition $\rho(x) = 1$ when $|x| < r$, and r is the maximum support radius. The effective support radius of the kernel is actually controlled by $\sigma(x)$, which is a decreasing function with respect to $|\nabla f(x_0)|$ – the local gradient strength. It is not difficult to see the drawback associated with this approach. Because the effective support radius is governed only by the local gradient strength, the filtering process is practically brought to a halt when an edge is encountered. A better approach would be to shape the kernel so that it is only narrow in the predominant gradient direction and a reasonable effective support radius is maintained to smooth along but not across the edge. In this case, the shape of the kernel becomes elliptical as shown in Fig. 1, where the axis that lies along the edge is called the *principal* (major) axis.

To accommodate this anisotropic nature of the filtering process, the following modification to Eq. (3) is necessary:

$$k(x_0, x) = \rho(x - x_0) e^{-\{((x - x_0) \cdot n)^2 / \sigma_1^2(x_0) + ((x - x_0) \cdot n_{\perp})^2 / \sigma_2^2(x_0)\}} \quad (4)$$

where n and n_{\perp} are mutually normal unit vectors, and n is in parallel with the principal axis. In Eq. (4), $(x - x_0) \cdot n$ and $(x - x_0) \cdot n_{\perp}$ are the dot products between vectors $(x - x_0)$ and n or n_{\perp} , respectively. The shape of the kernel is controlled through two non-negative functions $\sigma_1(x)$ and $\sigma_2(x)$. A relationship of $\sigma_1^2(x) + \sigma_2^2(x) = \sigma^2(x)$ and $\sigma_1(x) \geq \sigma_2(x)$ is usually maintained so that $k(x_0, x)$ is

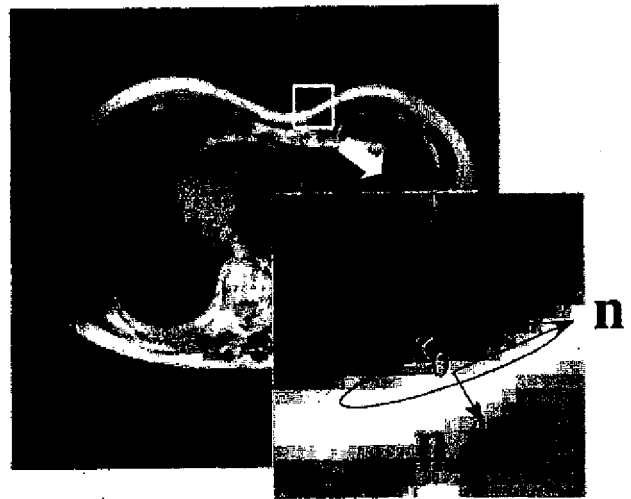


Fig. 1. Shaping of the filter kernel. When an edge is encountered the kernel is deformed into an ellipse with its principal axis aligned in parallel with the edge, and the smoothing is performed along but not across the edge.

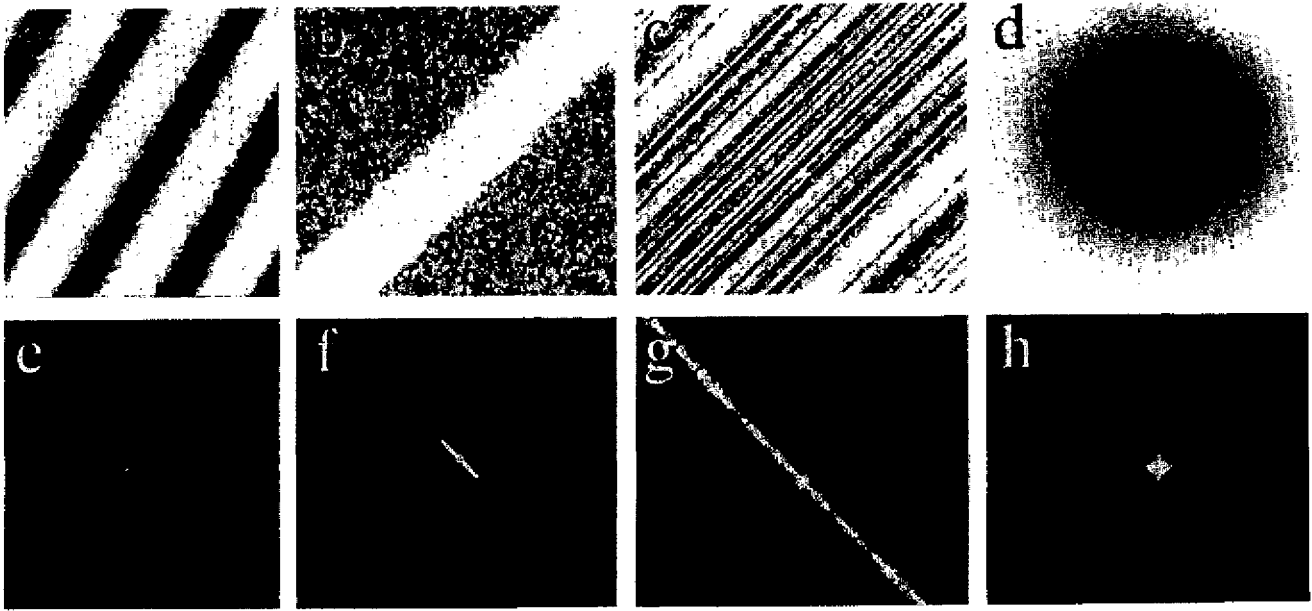


Fig. 2. Several examples of orientated patterns (a)–(c) and their corresponding Fourier transform (e)–(g). They demonstrate that the power spectrum of such patterns clusters along a line through the origin in Fourier domain. The direction of the line is perpendicular to the dominant spatial orientation. For a non-orientated pattern (d), the power spectrum (h) spreads out in all directions.

elongated along \mathbf{n} . One may notice that compared with Eq. (3), the constraints imposed on $\sigma_1(\mathbf{x})$ and $\sigma_2(\mathbf{x})$ in Eq. (4) have been loosened. They are no longer governed directly by the local gradient strength and, as we shall see later, they are also influenced by corners and other local image features.

To use Eq. (4) we have to define the direction of vectors \mathbf{n} and \mathbf{n}_\perp . The solution appears to be simple at a first thought, as one can define \mathbf{n} to be perpendicular to the local gradient direction. And, indeed, it was the approach that was adopted initially. Problems occur when the signal-to-noise ratio of the image is low and its derived local gradient vectors are noisy. To circumvent this problem, one can convolve the image with a 2D Gaussian kernel prior to calculating local intensity gradients. However, the smaller the signal-to-noise ratio, the larger the convolution window should be used. In this case, one is unintentionally assuming that the original signal varies smoothly within the selected window, and this assumption inevitably introduces errors when fine image details are encountered. If the orientation of the kernel is not properly aligned for anisotropic filtering, the image produced could, in fact, be worse than that of using a simple linear filtering technique. This motivated us to search for a better way of defining the filter kernel.

3. Local orientation detection

The basic idea behind our chosen technique for calculating the direction of \mathbf{n} was inspired by the work of

Bigün and Granlund [10] and that of Kass and Witkin [11]. The level contours (isogrey values) of an *orientated pattern* consist of parallel lines. This definition for an orientated pattern includes not only different forms of edges (step, slope and second order edges, etc.), but also patterns, as shown in Fig. 2. The power spectrum of such a pattern clusters along a line through the origin in the Fourier domain, and the direction of the line is perpendicular to the dominant spatial orientation. If we relate the infinitesimal energy $|F(\mathbf{u})|^2 d\mathbf{u}$ of the Fourier transform to a mass distribution, then the second moment (the moment of inertia of $|F(\mathbf{u})|^2 d\mathbf{u}$), $E(\mathbf{n}_p)$, with respect to a line through the origin with a unit normal vector \mathbf{n}_p is:

$$E(\mathbf{n}_p) = \int \dots \int d_p^2(\mathbf{u}) |F(\mathbf{u})|^2 d\mathbf{u} \quad (5)$$

where $d_p(\mathbf{u})$ is a real valued function which gives the shortest Euclidean distance between a point \mathbf{u} and a line that is perpendicular to \mathbf{n}_p , i.e.:

$$d_p^2(\mathbf{u}) = (\mathbf{u} \cdot \mathbf{n}_p)^2 = (\mathbf{n}_p^T \mathbf{u})(\mathbf{u}^T \mathbf{n}_p) = \mathbf{n}_p^T (\mathbf{u} \mathbf{u}^T) \mathbf{n}_p \quad (6)$$

where \mathbf{u} is the dual of \mathbf{x} in Fourier domain and T stands for matrix transpose. The above description is schematically illustrated in Fig. 3, in which the shaded area represents where the power spectrum of an orientated pattern is located.

By substituting Eq. (6) into Eq. (5) we have:

$$E(\mathbf{n}_p) = \int \dots \int \mathbf{n}_p^T (\mathbf{u} \mathbf{u}^T) \mathbf{n}_p |F(\mathbf{u})|^2 d\mathbf{u} = \mathbf{n}_p^T \mathbf{R} \mathbf{n}_p \quad (7)$$

where \mathbf{R} is a $m \times m$ matrix ($m = 2$ for 2D patterns, and 3

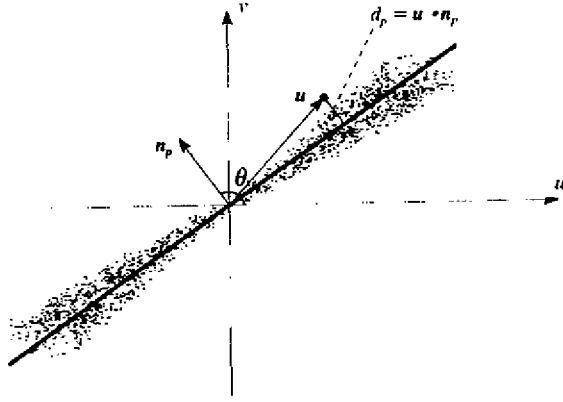


Fig. 3. A schematic diagram showing the distribution of the power spectrum of an orientated pattern and the derivation of its orientation direction. Vector n_p shown in the figure is the unit normal vector of a line that the power spectrum clusters along, whereas $d_p(u)$ is the shortest Euclidean distance between a point u and the line.

for 3D patterns), given by:

$$R = \int \dots \int (uu^T) |F(u)|^2 du \quad (8)$$

Matrix R will be referred to as the *second moment matrix* of the Fourier spectrum of image $f(x)$. The problem of finding the line that the power spectrum clusters along is now equivalent to finding n_p that minimizes function E .

For simplicity, we shall first look at orientated 2D patterns. Denote u_i ($i = 1, 2$) as components of u projected onto the two orthogonal axes; we have:

$$R_{ij} = \int \int u_i u_j |F(u_1, u_2)|^2 du_1 du_2 \quad (i, j = 1, 2) \quad (9)$$

By applying the power theorem and the fact that differentiation in the spatial domain corresponds to multiplication by the respective coordinate in the Fourier domain [12], Eq. (9) can be simplified and calculated directly in the spatial domain [13]; i.e.:

$$R_{ij} = \frac{1}{4\pi^2} \int \int \left(\frac{\partial f}{\partial x_i} \right) \left(\frac{\partial f}{\partial x_j} \right) dx_1 dx_2 \quad (10)$$

thus, no Fourier transform is involved in the actual calculation. Furthermore, function $E(n_p)$ defined in Eq. (7) is non-negative. Thus:

$$n_p^T R n_p = (n_p, R n_p) \geq 0, \quad \forall n_p \quad (11)$$

Eq. (11) states that R is positive semi-definite. From Eq. (10) it can be shown that R is also real symmetric, thus the minimization problem formulated in Eq. (7) is a matrix eigenvalue problem and it is solved by n_p , the vector that is parallel to the eigenvector corresponding to the smallest eigenvalue of the second moment matrix R . Since R is real symmetric and semi-definite, all eigenvalues of R are real and non-negative. The smallest eigenvalue λ_{\min} corresponds to the smallest value that

function $E(n_p)$ can attain, and the eigenvector that corresponds to λ_{\min} determines the estimated orientation direction.

Similarly, $\max(E(n_p)) = \lambda_{\max}$, where λ_{\max} is the maximum eigenvalue of R . This suggests the use of the following equation as a measure of anisotropy:

$$g(x) = \left\{ \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right\}^2 \quad (12)$$

For a pattern that is strongly orientated, $\lambda_{\max} \gg \lambda_{\min}$, thus $g \approx 1$, whereas for an isotropic pattern $\lambda_{\max} \approx \lambda_{\min}$ and $g \approx 0$. Analytically, the direction of n_p is calculated by:

$$\theta(x) = \frac{1}{2} \tan^{-1} \left\{ \frac{\int \int_{\Omega} 2 \left(\frac{\partial f}{\partial x} \right) \left(\frac{\partial f}{\partial y} \right) dx dy}{\int \int_{\Omega} \left(\left(\frac{\partial f}{\partial x} \right)^2 - \left(\frac{\partial f}{\partial y} \right)^2 \right) dx dy} \right\} + \frac{\pi}{2} \quad (13)$$

and the anisotropic measure is given by

$$g(x) = \frac{\left\{ \int \int_{\Omega} \left(\frac{\partial f}{\partial x} \right)^2 - \left(\frac{\partial f}{\partial y} \right)^2 dx dy \right\}^2 + \left\{ \int \int_{\Omega} 2 \left(\frac{\partial f}{\partial x} \right) \left(\frac{\partial f}{\partial y} \right) dx dy \right\}^2}{\left\{ \int \int_{\Omega} \left(\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) dx dy \right\}^2} \quad (14)$$

both of which can be calculated directly from the original data $f(x, y)$ and its partial derivatives. In Eqs (13) and (14) we have substituted x_1 and x_2 with x and y , respectively, and Ω is a local neighbourhood of $x = (x, y)$.

To demonstrate the effectiveness of the anisotropic measure stated by Eq. (14) and how it is used to control the shape of the filter kernel, a test pattern shown in Fig. 4a has been generated. This pattern has a varying anisotropic strength along the x direction. Fig. 4c shows the same pattern but with some additive noise. The results of applying Eq. (14) to Figs 4a and 4c are given in Figs. 4b and 4d, respectively. A fixed window of 7×7 pixels was used for the calculation. It was positioned in the vertical centre of the image and shifted horizontally from left to right. As can be seen from the figures, the anisotropic measure remains high for an image that is noise free. Otherwise, $g(x)$ drops as the anisotropic strength decreases thus makes the effective support radius increase along the horizontal direction. For pixels on the left side of the image, a narrower kernel is used to avoid destroying image structure during the filtering process. On the other hand, a wider kernel is adopted for pixels towards the right as the signal varies smoothly, a higher efficiency thus can be achieved as the kernel now covers a larger area. This adaptiveness to local image structure ensures that the *efficiency* and *fidelity* conditions mentioned earlier are satisfied throughout the filtering process.

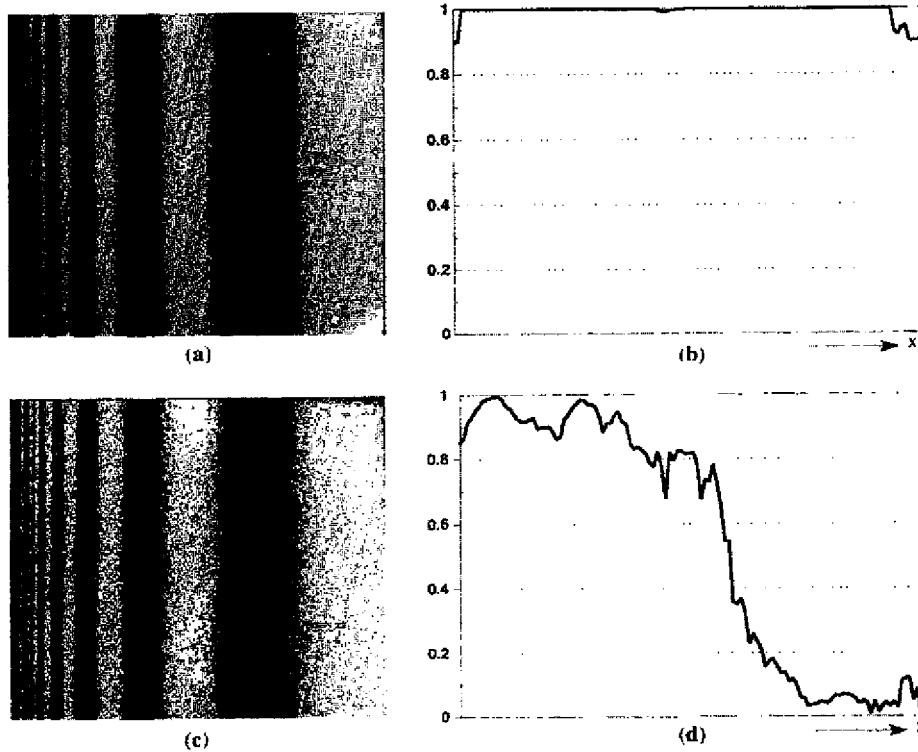


Fig. 4. A synthetic orientated pattern (a) with its anisotropic strength (b) measured along the x axis. The same pattern with additive noise (c) and its measured anisotropic strength (d).

4. Corner recognition

The anisotropic measure defined above not only gives the indication of how strong a pattern is orientated, but also provides a convenient way of finding corner and junction points within a given image.

It is known that at edges the anisotropic measure $g(x)$ is close to one, thus $\psi(g(x)) = 1 - g(x) \approx 0$. On the other hand, at flat regions the gradient strength $|\nabla f(x)|$ is close to zero. It is only at corner and junction points that $1 - g(x) \gg 0$ and $|\nabla f(x)| \gg 0$. This suggests the use of the following expression as a measure of *corner strength*:

$$c(x) = \psi(g(x))|\nabla f(x)|^2 = (1 - g(x))|\nabla f(x)|^2. \quad (15)$$

In Eq. (15), $\psi(g(x))$ can generally be chosen as a monotonically decreasing function with respect to $g(x)$ as long as it satisfies:

$$\psi(g(x)) \in [0, 1] \quad \text{and} \quad \psi(g(x)) = \begin{cases} 0, & \text{when } g(x) = 1 \\ 1, & \text{when } g(x) = 0 \end{cases} \quad (16)$$

The method stated above involves minimum extra computation, and we shall see in the next section that our results acquired compare favourably with other popular corner detection techniques. For example, one well known method relies on the derivative of the

gradient directions (the curvature of level contours) [14] and the equation derived is in the form of:

$$c(x) = \frac{\frac{\partial^2 f}{\partial y^2} \left(\frac{\partial f}{\partial x} \right) - 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right) \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial x^2} \left(\frac{\partial f}{\partial y} \right)}{\left\{ \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right\}^{3/2}} \quad (17)$$

To obtain a realistic indication of corner strength, the popular Kitchen-Rosenfeld corner detector [15] is usually employed which, in effect, is a multiplication of the above measure with the local gradient strength, i.e.:

$$c(x) = \frac{\frac{\partial^2 f}{\partial y^2} \left(\frac{\partial f}{\partial x} \right) - 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right) \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial x^2} \left(\frac{\partial f}{\partial y} \right)}{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2} \quad (18)$$

This involves second order derivatives of the original image, and problems may arise if the input data is noisy. This is because taking derivatives is a high-pass filtering process which amplifies the effect of noise. Other available techniques are based on calculating the histogram of local directional derivatives which entails an

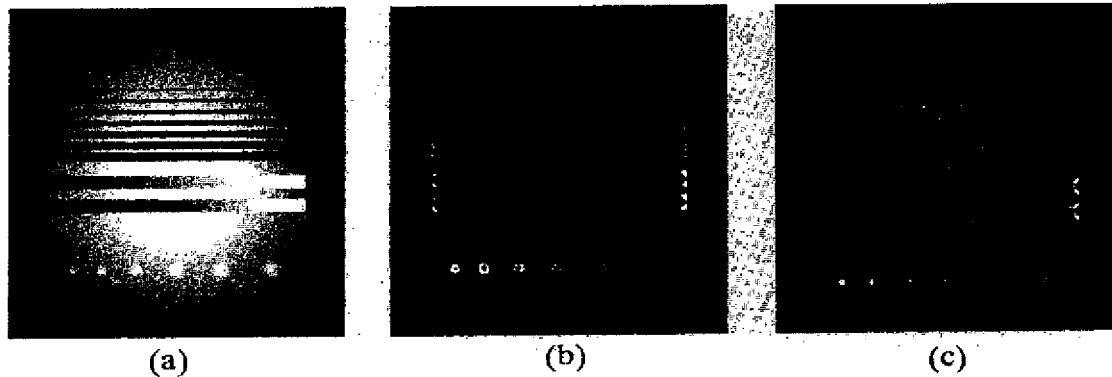


Fig. 5. A synthetic image with additive noise (a) and the filtered image (b) after applying the proposed corner detector. Figure (c) is the result of using the Kitchen-Rosenfeld corner detector as described by Eq. (18) on image (a).

analysis of the following different cases:

- *Twin peaks* separated by 180° — straight lines or edges;
- *Twin peaks* separated by 90° or less — right angle or sharp corners;
- *Single peak* — termination points for lines;
- *Multiple peaks* — junctions.

Yet again, this method is sensitive to noise and a stable result is not always easy to achieve in practice. Furthermore, it involves a high computational cost as it requires histogram peak searching for each local window.

By incorporating the corner strength $c(x)$ into Eq. (4), the proposed adaptive filtering scheme can finally be formalized as:

$$\sigma_1(x_0) = \frac{r}{1 + c(x_0)/\alpha}, \quad \text{and} \quad \sigma_2(x_0) = (1 - g(x_0)) \frac{r}{1 + c(x_0)/\alpha} \quad (19)$$

in which α is a normalization factor that controls how faithfully the corners and junctions should be preserved during the filtering process. It is usually selected as a

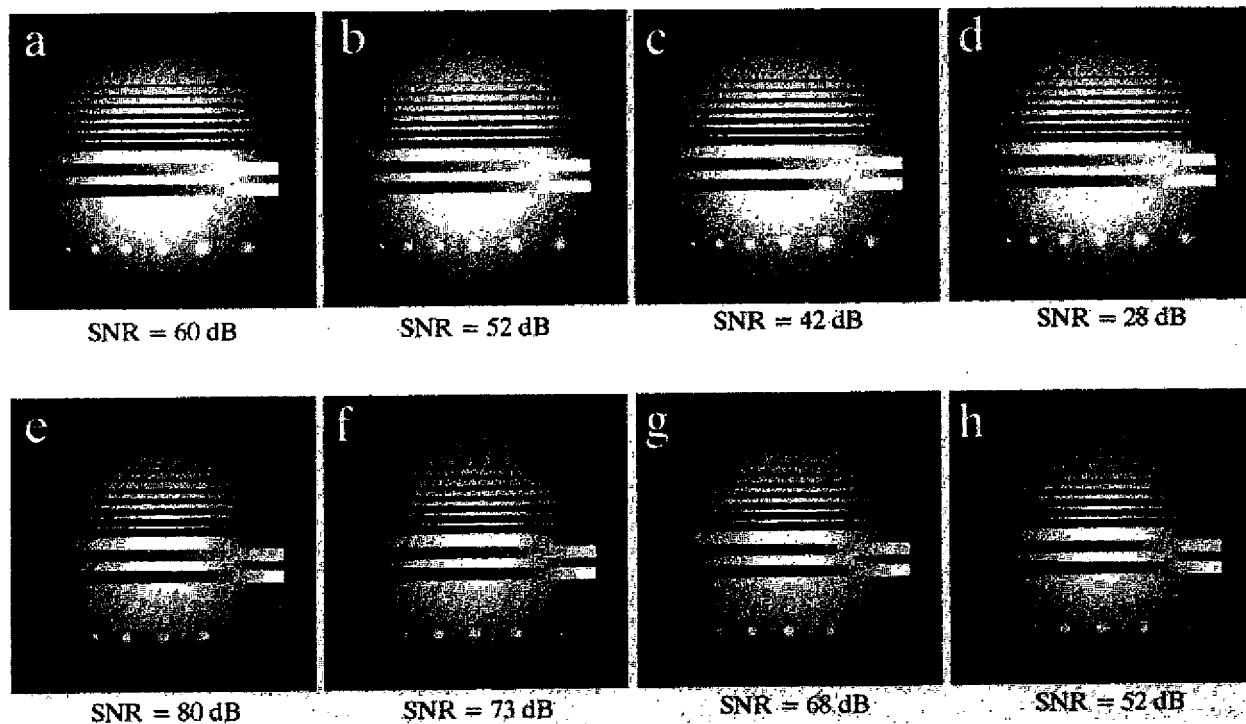


Fig. 6. A synthetic image with different levels of additive noise (a)–(d) and the corresponding filtered images (e)–(h). The thin line at the top of the image has a width of only one pixel, and it starts to fade away when the signal-to-noise ratio decreases. The preservation of edge is good whereas a slight degradation of corners can be observed as SNR decreases.

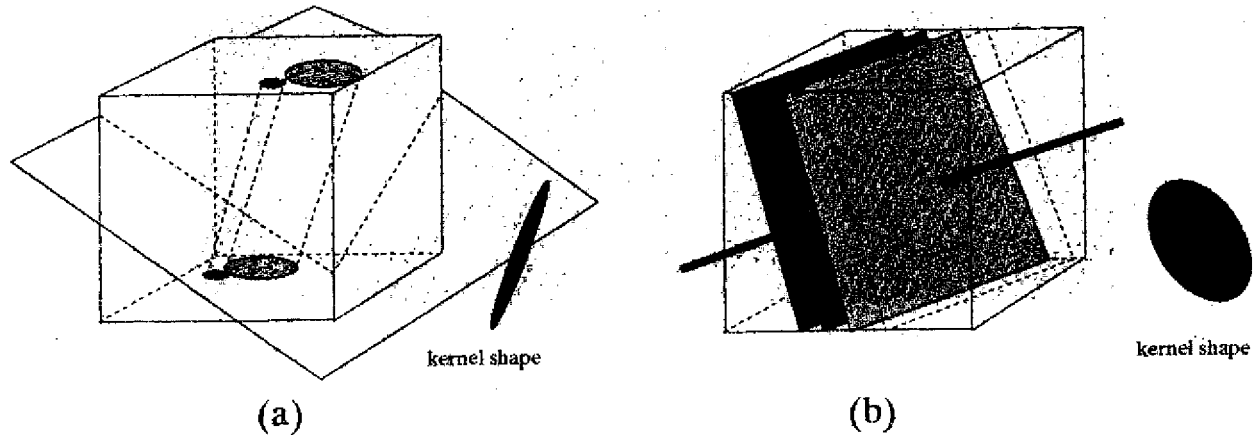


Fig. 7. (a) A first order orientated pattern with its Fourier spectrum clustering within a plane that is perpendicular to the orientation direction. The shape of the filter is like a thin rod tapering at both ends. (b) A second order oriented pattern with its Fourier spectrum clustering along a line. The minimum eigenvalue of the second moment matrix R has a multiplicity of two. The eigenvector corresponding to the maximum eigenvalue is in parallel with the line that the power spectrum clusters around. The shape of the filter kernel is this case is like a thin disk which is parallel to the orientation of iso-surface layers.

value between 50% and 200% of the maximum corner strength within the image. In Eq. (19) r is the maximum support radius, as mentioned earlier. It can be seen that as the corner strength $c(x)$ increases, both $\sigma_1(x_0)$ and $\sigma_2(x_0)$ decrease to ensure that a small effective window is used to avoid smearing. The shape of the kernel, on the other hand, is adjusted by the term $(1 - g(x_0))$ in the expression for $\sigma_2(x_0)$. When an edge is encountered the term $(1 - g(x_0))$ is small. The extent of the kernel is thus deformed from a circle to an ellipse, which guarantees that the smoothing is performed predominantly along but not across the edge.

5. Results for synthetic images

The proposed technique has been verified with a synthetic image. Fig. 5a shows a synthetic image with additive noise and features a group of lines with different width, a first and a second order grey level strip (the increase in pixel intensities along the horizontal direction is proportional to x and x^2 , respectively) and five disks with different sizes and intensities. All these are embedded into a large disk with varying intensity from the centre outwards. It is constructed in this way so that the results will reflect the filter response with respect to different image details. Fig. 5b shows the corner strength $c(x)$ when Eq. (15) is applied. As a comparison, Fig. 5c gives the result of detected corners by the Kitchen-Rosenfeld operator. The contrast between Fig. 5b and 5c is obvious. Fig. 5c not only has higher background noise, but also gives a lot of false responses. Some of the lines are also detected as corners because they are embedded in a background with varying pixel intensities. Our proposed method clearly handles this situation much better. As a valuable by-product derived from

this work, the corner detection scheme mentioned here can be a useful tool for other image recognition and computer vision applications.

Figs. 6a–d show the synthetic image with different signal-to-noise ratios, i.e. 60 dB, 52 dB, 42 dB and 28 dB, respectively. The measure of the signal-to-noise ratio is defined by:

$$SNR = 10 \log_{10} \frac{\int \dots \int s^2(x) dx}{\int \dots \int \gamma^2(x) dx} \quad (20)$$

with $s(x)$ and $\gamma(x)$ denoting the value of the original image and the additive noise, respectively. The input image to the filter is $f(x) = s(x) + \gamma(x)$.

Figs. 6e–h are the processed results of Figs. 6a–d, respectively. The thin line at the top of the synthetic image has a width of only one pixel, and it can be seen that it starts to fade away when the SNR decreases. The preservation of edges is generally good, though a slight degradation of corners is noticeable. The maximum support radius used was 3 pixels, and the value α was set to be 75% of the maximum of $c(x)$ within the image. An increase of more than 20 dB in SNR has been achieved with very minor degradation of local features.

6. Considerations for 3D implementation

The extension of the technique to 3D requires some careful consideration. A 3D image can be considered as a 2D cine sequence, or a 3D volume data set. There exist two situations for 3D orientated patterns: one is that the isogrey values of the intensity pattern in 3D space

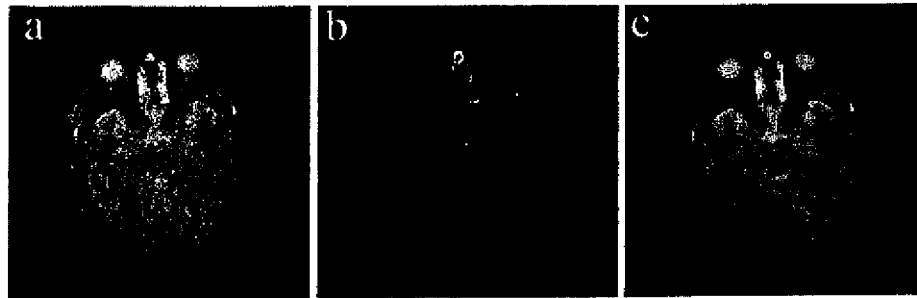


Fig. 8. A single trans-axial head image (a) with its detected corner strength (b) and the filtered image (c). The maximum support radius is 3 pixels, thus giving a window size of 7×7 pixels.

constitute parallel *lines*; and the other is that the isogrey values form parallel *planes*. These two situations are illustrated in Figs. 7a and 7b, respectively. The former corresponds to conditions in MR imaging where small vessels run in parallel within a local area (e.g. peripheral blood vessels shown in MR angiograms), whereas the latter corresponds to thin surface layers lying in parallel with each other. These two cases are termed as the *first* and *second order* orientated patterns, respectively. For first order orientated patterns, the Fourier spectrum clusters along a plane through the origin with its normal defined by the eigenvector that corresponds to the smallest eigenvalue of the second moment matrix R . For second order orientated patterns, the Fourier spectrum of $f(x)$ shrinks into a line through the origin.

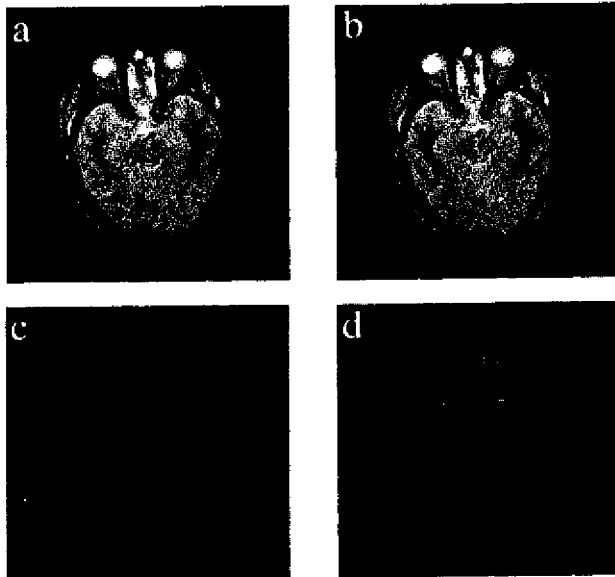


Fig. 9. (a)–(b) Filtered results for the image shown in Fig. 8(a). The maximum support radius used for (a) was 4, whereas for (b) was 2. The results are very similar to Fig. 8(c). (c)–(d) The differences between 9(a), 9(b) and Fig. 8(c). The images shown in (c) and (d) have been amplified 40 times and are displayed with the same brightness and contrast settings as those for Figs. 9(a) and 9(b).

The smallest eigenvalue of R in this case has a multiplicity of two, thus for any plane that contains the line the function $E(n_p)$ will be minimized. The sub-space defined by the normal vectors of these planes corresponds to a plane that is defined by two orthonormal eigenvectors of λ_{\min} . Since all the eigenvectors of R are orthonormal with respect to each other, the eigenvector that corresponds to λ_{\max} will be parallel to the line around which the power spectrum clusters.

The extension of Eq. (4) to accommodate 3D orientated patterns can be subsequently described by the following expressions:

$$k(x_0, x) = \rho(x - x_0) e^{-\sum_{i=1}^3 ((x-x_0) \cdot n_{\lambda_i})^2 / \sigma_i^2(x_0)} \quad (21)$$

where n_{λ_i} is the eigenvector that corresponds to λ_i , and:

$$\sigma_i(x_0) = \left\{ 1 - \left(\frac{\lambda_i - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 \right\} \frac{r}{1 + c(x_0)/\alpha}, \quad (i = 1, 2, 3) \quad (22)$$

with $c(x)$ as has been defined in Eq. (15). As in Eq. (19), α is a normalization factor and r is the maximum support radius. They are combined with the corner strength $c(x)$ to regulate the effective window size. The three eigenvalues are used here to control the shape of the filter kernel. As shown in Fig. 7, for first order orientated patterns, there are one small and two large eigenvalues, and the kernel has the shape of a thin rod tapering at both ends, whereas for second order orientated patterns there are one large and two small eigenvalues; consequently, the kernel appears like a disk which is parallel to isogrey level surfaces.

7. Application results and discussion

Several different types of MR images have been used to evaluate the performance of the proposed algorithm for noise reduction. These images were acquired with a 0.5T MR scanner (Royal Brompton, London). All images in the experiments have the same data array

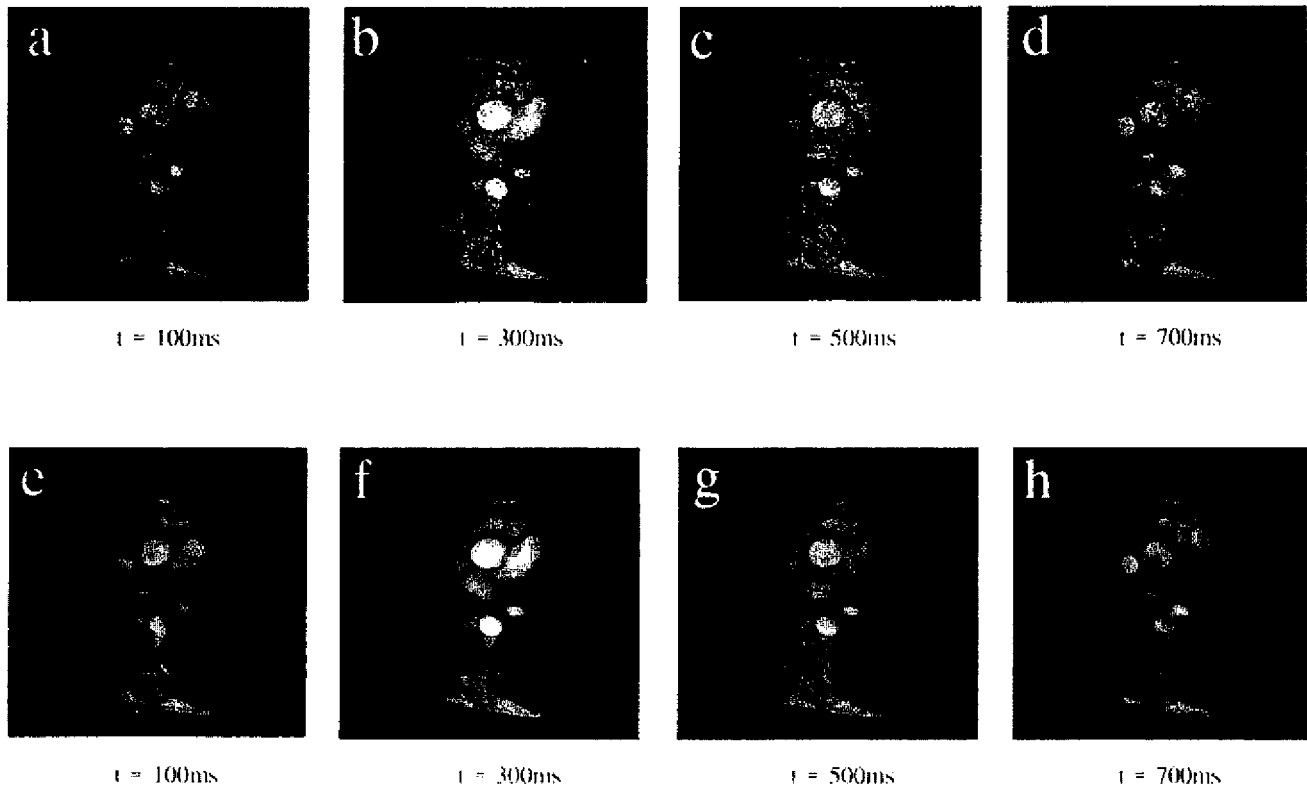


Fig. 10. Four frames (a)–(d) of a gradient echo cine oblique view of the ascending and descending aorta with their corresponding filtered results (e)–(h). The improved region definition facilitates the region segmentation required for aortic compliance studies.

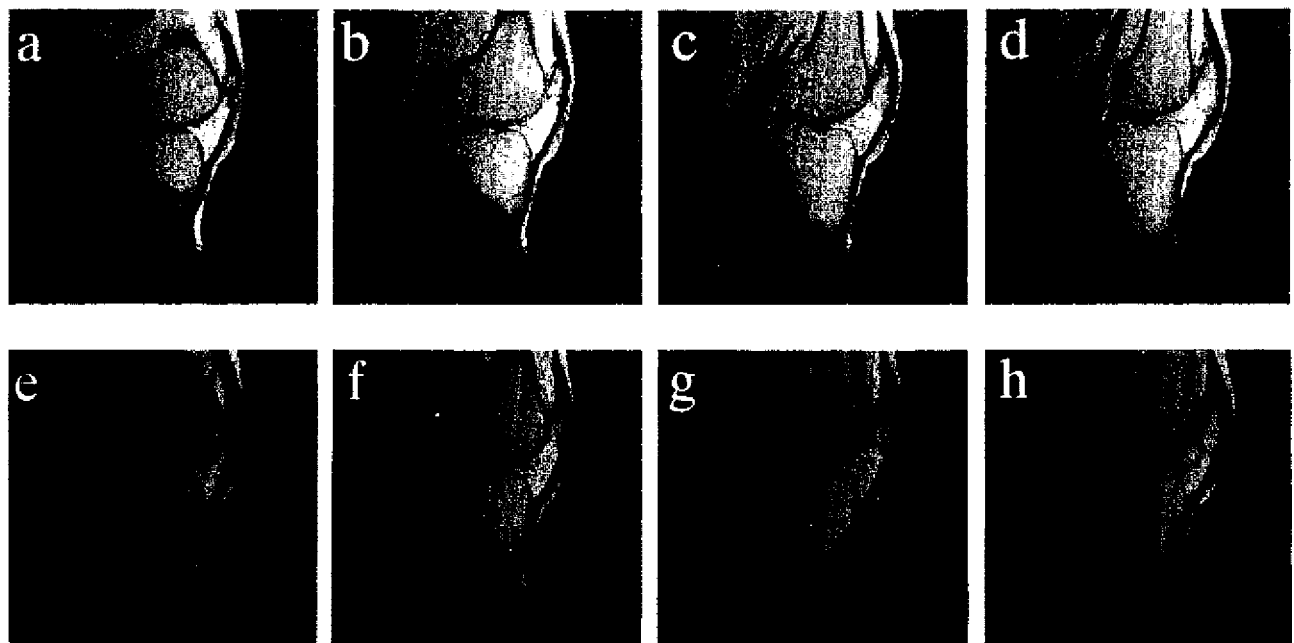


Fig. 11. Results of applying the proposed algorithm to a multi-slice spin-echo acquisition of a knee. The top row (a)–(d) shows the original four of the 15 slices and the bottom row (e)–(h) gives the corresponding filtered results.

size of 256×256 with 1 byte/pixel. They were processed on a Silicon Graphics IRIS Indigo R4000 workstation.

In the first experiment a single trans-axial head image from a two-dimensional acquisition is used. The original and the filtered images are shown in Figs. 8a and 8c, respectively. Fig. 8b is the detected corner strength which controls the extent of the filter kernel. The improvement of visual quality and structure definition is evident. The maximum support radius used in this experiment was 3, thus giving a window size of 7×7 pixels. As a comparison, Figs. 9a and 9b provide results with $r = 2$ and $r = 4$, respectively. The differences between Figs. 9a, 9b and Fig. 8c are given in Figs. 9c and 9d, respectively, where the subtracted result has been amplified 40 times to reveal small discrepancies. It appears that the final filtered images do not rely heavily on r . This is a desirable feature for images with coarse as well as fine structures. The value α for controlling how well the corners and junctions are preserved was set to be the same as in previous calculations, i.e. 75% of the maximum of $c(x)$ within the image.

Our second experiment used a gradient echo cine oblique view of the ascending and descending aorta. Sixteen frames were acquired within each cardiac cycle, and the 3D version of the algorithm has been applied to the data set. Only four images are shown here. They were gated at 100 ms, 300 ms, 500 ms and 700 ms after the onset of the ECG (electrocardiogram) *R* wave. The filtered images are shown in Figs. 10e–h. These results show improved region definition which facilitates the region segmentation required for aortic compliance studies.

As the last example, we applied this algorithm to a multi-slice spin-echo acquisition of the knee. The images are of poor quality due to the thin slice and inferior MR receiving coils used. The structures in this data sets have many thin muscle surface layers. Using the algorithm with the same parameter settings as before, we obtained the enhanced images as shown in Fig. 11. There is no obvious loss of structural information, and an improvement of both visual quality and structure definition are apparent.

It is worth commenting on techniques that rely on the anisotropic diffusion scheme proposed by Perona and Malik [8]. As has been discussed earlier, the method allows an image $f(x)$ to evolve over time via the diffusion equation:

$$\frac{\partial f(x, t)}{\partial t} = D(x, t) \Delta f(x) + \nabla D \cdot \nabla f(x) \quad (23)$$

where $D > 0$ is a decreasing function with respect to the edge strength and Δ is the Laplacian operator. The result is to diffuse $f(x)$ most where the gradient is smallest and least where the gradient is largest. Since the orientation of the edge has not been taken into account, the efficiency of the diffusion process is not ideal. When an input image

contains a considerable amount of noise, the resulting random fluctuations of $|\nabla f(x)|$ slow down the diffusion process. In our experience, we noticed that the filtered result can look patchy when very noisy images are encountered. Another major drawback of the technique is that it distorts sloping edges. As has been shown [16], an edge can deform into a multiple or single jagged one with a shape that is dependant on selected parameter values. Furthermore, the filtered results converge ultimately to a piecewise constant image definition, thus only a small number of iterations should be used if subtle image details are to be preserved. The method described in this paper, however, does not have all these limiting factors. It does not affect the image fidelity and can at the same time keep up with the efficiency in terms of noise filtering. In addition, all parameter values used proved to be robust, and this makes the algorithm easily applicable to images with varying structures.

8. Conclusions

The technique proposed in this paper relies on the detection of local orientation features by adaptively controlling the filter kernel. The detection of local orientation is based on the Fourier analysis of the intensity pattern. The evaluation of Fourier transforms, however, is not necessary for the actual calculations. A simple relationship between the local orientation direction and matrix eigenvectors has been obtained. The eigenvalues and their corresponding eigenvectors are used to control the shape and the extent of the filter kernel. Application of this filtering process ensures that corners and junctions are well preserved. Experimental results obtained by the application of our method to both synthetic and real images show that the algorithm is robust and can cope with low signal-to-noise ratio. The noise filtering efficiency of the algorithm is good; the computational efficiency, however, still needs improvement. The current execution time for a 256×256 image with 1 byte/pixel is about 12–18 seconds, depending on the size of the maximum support radius. We are currently implementing a parallel algorithm to improve the efficiency of our method.

Acknowledgements

The authors would like to thank CORDA (the coronary artery disease research association) for its financial support, and colleagues in the Magnetic Resonance Unit of Royal Brompton Hospital for their sound advice and critical suggestions.

List of symbols

x	position vector of a point in space
x_i	the i th component of x
u	dual of x in Fourier domain
u_i	the i th component of u
x_0	a given point in space
n	a unit normal vector to a line (in 2D) or plane (in 3D)
n_p	a unit normal vector to a candidate line (in 2D) or plane (in 3D) that the power spectrum of an orientated pattern clusters along
n_{\perp}	a unit vector that is perpendicular to n
Ω	local neighbourhood of x_0
$s(\cdot)$	the original image (not corrupted by noise)
$\gamma(\cdot)$	additive noise to $s(\cdot)$
$f(\cdot)$	input image, $f(\cdot) = s(\cdot) + \gamma(\cdot)$
$f_s(\cdot)$	filtered result of $f(\cdot)$
$F(\cdot)$	the Fourier transform of $f(\cdot)$
$k(\cdot)$	filter kernel
$\mu(\cdot)$	a normalization factor for $k(\cdot)$
$\rho(\cdot)$	a positive and rotationally symmetric cut-off function
r	the maximum support radius of $k(\cdot)$
α	a normalization factor for adjusting the influence of corner strength in $k(\cdot)$
$\sigma_1(\cdot)$	a function that controls the major axis of the filter kernel
$\sigma_2(\cdot)$	a function that controls the minor axis of the filter kernel
$d_p(u)$	the shortest Euclidean distance between a point u and a line that has normal vector of n_p
R	the second moment matrix of the Fourier spectrum of image $f(x)$
R_{ij}	the (i, j) th element of matrix R
λ	eigenvalue of R
λ_i	the i th eigenvalue of R
λ_{\min}	smallest eigenvalue of R
λ_{\max}	largest eigenvalue of R
$g(\cdot)$	the local anisotropic measure of an orientated pattern
$c(\cdot)$	corner strength measure
$\psi(g(x))$	a decreasing function with respect to $g(x)$ used for composing $c(\cdot)$
$E(n_p)$	the moment of inertia of the Fourier spectrum with respect to a line (or plane in 3D) through the origin with a unit normal vector n_p

D	diffusion parameter
T	matrix transpose
Δ	Laplacian operator
∇	gradient operator
$ \cdot $	magnitude of a complex number or a vector
∂	partial derivative operator

References

- [1] J.S. Lee, Digital image enhancement and noise filtering by use of local statistics, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(2) (1980) 165–168.
- [2] D.C. Wang, A.H. Vagucci and C.C. Li, Gradient inverse weighted scheme and the evaluation of its performance, *Computer Graphics and Image Processing*, 15 (1981) 167–181.
- [3] X. Wang, On the gradient inverse weighted filter, *IEEE Trans. Signal Processing*, 40(2) (1992) 482–484.
- [4] G.A. Mastin, Adaptive filters for digital image noise smoothing: an evaluation, *Computer Vision, Graphics, and Image Processing*, 31 (1985) 103–121.
- [5] A. Lev, S.W. Zucker and A. Rosenfeld, Iterative enhancement of noisy images, *IEEE Trans. Systems, Man, and Cybernetics*, 7 (1977) 435–441.
- [6] L.S. Davis and A. Rosenfeld, Noise cleaning by iterated local averaging, *IEEE Trans. Systems, Man and Cybernetics*, 8 (1978) 705–710.
- [7] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.
- [8] P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12 (1990) 629–639.
- [9] G. Gerig, O. Kübler, R. Kikinis and F.A. Jolesz, Nonlinear anisotropic filtering of MRI data, *IEEE Trans. Medical Imaging*, 11 (1992) 221–232.
- [10] J. Bigün and H. Granlund, Optimal orientation detection of linear symmetry, *Proc. First International Conference on Computer Vision, London*, 1987, pp. 433–438.
- [11] M. Kass and A. Witkin, Analysing oriented patterns, *Computer Vision, Graphics and Image Processing*, 37 (1987) 362–397.
- [12] R. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, New York, 1986, 2nd revised edn.
- [13] G.Z. Yang and P. Burger, Optimal extraction of image flow from spatio-temporal images, *Imperial College Research Report*, DOC 90/9, 1990, University of London.
- [14] M. Kass, A. Witkin and D. Terzopoulos, Snakes: active contour models, *Proc. First International Conference on Computer Vision, London*, 1987, pp. 259–268.
- [15] L. Kitchen and A. Rosenfeld, Gray level corner detection, *Pattern Recognition Letters*, 1 (1982) 95–102.
- [16] M. Nitzberg and T. Shiota, Nonlinear image filtering with edge and corner enhancement, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14 (1992) 826–830.

Morphological Corner Detection

Robert Laganière

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ont.

CANADA K1N 6N5

Abstract

This paper presents a new operator for corner detection. This operator uses a variant of the morphological closing operator, which we have called asymmetrical closing. It consists of the successive application of different morphological transformations using different structuring elements. Each of these structuring elements used to probe the image under study is tuned to affect corners of different orientation and brightness. We found that this kind of approach, based on brightness comparisons, leads to better quality results than others and is achieved at a lower computational cost.

1 Introduction

Corners constitute attractive 2D features, often used in computer vision for tasks such as stereovision, 3D interpretation, motion estimation, and structure from motion. They abound in indoor scenes where several polyhedral objects and intersecting planes (floor, walls, etc.) are present. Corners serve as points of interest in two-view matching algorithms [1][2][3]. Corner detection is also used in camera calibration for the localization of reference points on a calibration pattern ([4] for example).

Corner detection is sometimes realized through the analysis of binary edge maps from which chain codes are extracted in order to find high curvature points [5][6][7]. However, most approaches work directly at the grayscale level [8]-[17]. These methods usually use local measurements in order to obtain a *corner strength*. Non-maxima suppression and thresholding lead then to a binary map showing where corners have been detected. These corner finder are usually characterized by an accuracy of few pixels and a relatively high level of false positives. Model-based approaches such as [18][19] also exist and allow corner localization at a subpixel accuracy. But these methods are more CPU-intensive and are only used after a first corner map has been obtained.

One of the difficulties with corner detection lies in

the corner definition itself. A restrictive description simply defines corners as the junction of two homogeneous regions separated by a high-curvature boundary. This definition is incomplete since it does not include X, Y and T junctions that should also be categorized as corners since they might be the image of 3D corners (intersection of three planes). A less rigid definition assimilates corners to points with high derivatives in several directions. This is a very loose description of the term *corner* since several “non-corner” points fall into this category.

This paper proposes an approach to corner detection based on mathematical morphology. The goal was to obtain a fast corner detector that is accurate, stable, selective and robust to noise. The next section is a short review of existing corner detectors. Section 3 presents some mathematical morphology concepts. Section 4 describes the proposed corner detector and Section 5 shows some comparative results. Section 6 is a conclusion.

2 Corner detection

We review here the main corner detectors that work directly at the grayscale level. All these methods use local measurements in order to obtain a corner strength $c(x, y)$ for each point of the image. Local non-maxima suppression and thresholding are then performed in order to extract points that will be reported as corners.

The use of the Hessian determinant of the intensity surface to estimate corner strength has been proposed by Beaudet [8]. Kitchen and Rosenfeld [9] proposed to use the gradient magnitude and the rate of change of gradient direction along an edge contour. Very similar operators have also been proposed by Dreschler and Nagel [10] and Zuniga and Haralick [11]. Deriche and Girondin [12] proposed a scale-based approach that uses the Beaudet's operator in conjunction with the Laplacian.

Following the idea of points of interest developed

by Moravec [13], the Plessey detector [14] is based on the following matrix:

$$\mathbf{M}(x, y) = \begin{bmatrix} \langle \left(\frac{\delta I(x, y)}{\delta x} \right)^2 \rangle & \langle \frac{\delta I(x, y)}{\delta x} \frac{\delta I(x, y)}{\delta y} \rangle \\ \langle \frac{\delta I(x, y)}{\delta x} \frac{\delta I(x, y)}{\delta y} \rangle & \langle \left(\frac{\delta I(x, y)}{\delta y} \right)^2 \rangle \end{bmatrix} \quad (1)$$

where $\langle \cdot \rangle$ denotes a smoothing operation on \mathbf{I} . Corner strength has been first defined by Noble [15] from which a slightly different version has been proposed by Harris and Stephens [16]:

$$c_{HS}(x, y) = \text{Det}(\mathbf{M}(x, y)) - k \text{Trace}^2(\mathbf{M}(x, y)) \quad (2)$$

The role of the parameter k is to remove sensitivity to strong edges.

All of the above methods are based on directional derivatives. They all suffer from the same drawback: local estimation of derivatives is very sensitive to noise and, when smoothing is applied, the corner localization precision is reduced. In addition, the computational complexity of the smoothing operation, derivative estimation and corner strength computation that are involved in such methods can be quite high.

A simpler approach based on brightness comparisons has been proposed by Smith and Brady [17]. The SUSAN corner detector is a modified version of the edge detector of the same name. It is based on the computation of the area of points inside a circular region \mathcal{N}_{xy} having a brightness similar to the one of the center point (x, y) . This area is computed as follows:

$$n(x, y) = \sum_{(i, j) \in \mathcal{N}_{xy}} e^{-\left(\frac{I(i, j) - I(x, y)}{t} \right)^6} \quad (3)$$

The parameter t controls the sensitivity to noise, i.e. it defines the similarity between brightness values. The value of $n(x, y)$ is therefore compared to a fixed threshold equal to $n_{max}/2$ where n_{max} is the maximum value that $n(\cdot)$ can take, that is:

$$c_S(x, y) = \begin{cases} \frac{n_{max}}{2} - n(x, y) & \text{if } n(x, y) < \frac{n_{max}}{2} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The value of this function corresponds to the corner strength. In order to reduce the number of false positives due to smooth boundary, thin lines and fine textures, two criteria must be added. The center of gravity of the circular region must be (1) located sufficiently far away from the center point and (2) all pixels lying in a straight line from the center point to the center of gravity must be of similar brightness.

Because of their computational simplicity, methods based on brightness comparisons constitute an attractive solution to the problem of corner detection. Moreover, our experiments showed that this kind of approach allied accurate corner localization with good robustness to noise. The corner detector we propose follows this approach and makes use of some basic morphological tools.

3 Mathematical morphology

Mathematical morphology is a methodology for image analysis that has been widely used in computer vision. The principle of all basic morphological operators is to probe the image under study with a *structuring element*. This structuring element is a set of pixels on which an origin is defined. To evaluate the results of a morphological operation on an image point, the structuring element is translated in such a way that its origin coincides with this image point. The shape of the structuring element defines a set $\mathbf{I}_{SE}(x, y)$ that includes all pixels of the image hit by the structuring element. From this set, the elementary morphological operators *erosion* and *dilation* can now be defined. The erosion of an image \mathbf{I} with a structuring element SE is given by:

$$\mathbf{I}_{SE}^e(x, y) = \min \mathbf{I}_{SE}(x, y) \quad (5)$$

Similarly, the dilation of an image \mathbf{I} with a structuring element SE is given by:

$$\mathbf{I}_{SE}^d(x, y) = \max \mathbf{I}_{SE}(x, y) \quad (6)$$

Two morphological transformations are defined by the successive application of these operators. The opening of an image \mathbf{I} by a structuring element SE is defined as an erosion followed by a dilation:

$$\mathbf{I}_{SE}^o = (\mathbf{I}_{SE}^e)^d_{\check{SE}} \quad (7)$$

where \check{SE} is the symmetrical transposition of SE with respect to its origin. All image structures that cannot contain the structuring element are removed by the opening. Therefore, the shape and size of the structuring element must be set according to the information that is to be extracted. The closing of an image \mathbf{I} by a structuring element SE is defined as a dilation followed by an erosion:

$$\mathbf{I}_{SE}^c = (\mathbf{I}_{SE}^d)^e_{SE} \quad (8)$$

4 Asymmetrical closing for corner detection

In the context of corner detection, one interesting choice is to consider a cross-shaped structuring element (Figure 1(a)). Indeed, because of the particular shape of this structuring element, the opening and

closing operators alter mainly this kind of image structure. However, these corner detectors suffer from three problems:

1. Opening affects only bright corners over dark background while closing affects only dark corners over bright background.
2. Small image structures (including impulsive noise and thin lines) are also eliminated and thus can be wrongly assimilated to corners.
3. This kind of corner detection is not rotationally invariant.

A concurrent application of opening and closing can solve the first problem, but we introduce a more efficient solution. We propose to perform what we call an *asymmetrical closing*, that is, the dilation of an image using a given structuring element followed by an erosion using another structuring element (note that asymmetrical opening could also have been considered). The central idea is to make dilation and erosion complementary in terms of the type of corners they affect. This can be realized by choosing a cross as the first structuring element and a lozenge for the second one (Figure 1(b)). We can then write the asymmetrical closing as:

$$I_{+, \diamond}^c = (I_+^\delta)_\diamond^e \quad (9)$$

and corner strength will be given by comparing the resulting image with the original one, that is:

$$c_+(\mathbf{I}) = |\mathbf{I} - \mathbf{I}_{+, \diamond}^c| \quad (10)$$

Basically, the value of $c(x, y)$ corresponds to the brightness difference between the corner and its background. Note that, as defined, the transformation produces a three-pixel L-shaped response in the case of dark corners. We also observed this kind of multiple-pixel response for real images. This must be interpreted as a consequence of the fact that the precise location of smooth corners is not well defined. If needed, it is still possible to select only the central point in each set. Corners detected by c_+ on a test image are shown in Figure 2(a) (for all experiments to follow, we used the structuring elements shown in Figure 1). Clearly, rotational invariance and small structure sensibility have not been solved. To detect the missing corners, the following operator can be used (which is a 45° rotated version of the preceding one):

$$c_\times(\mathbf{I}) = |\mathbf{I} - \mathbf{I}_{\times, \square}^c| \quad (11)$$

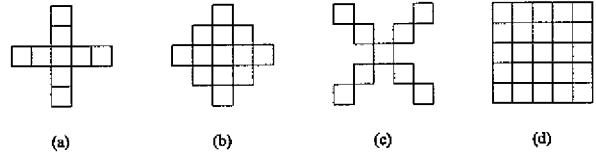


Figure 1: (a) structuring element +. (b) structuring element \diamond . (c) structuring element \times . (d) structuring element \square .

Figure 2(b) shows the corners detected by this operator. Surprisingly, it appears that the two complementary operators, c_+ and c_\times , are sufficient to detect corners of almost any orientation. While these two operators are sensitive to corners of different orientation, they are both sensitive to the same small structures. Consequently, the combination of these two operators should make corner detection almost rotationally invariant and insensitive to small image structures. This leads to the following operator:

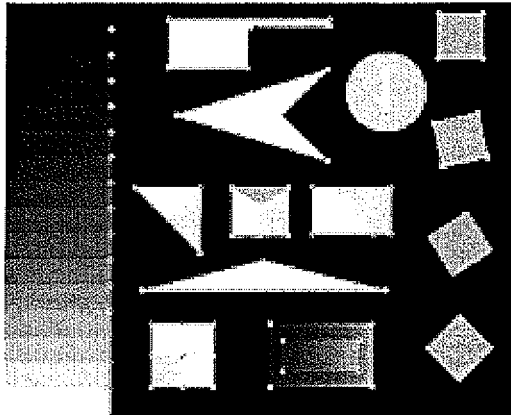
$$c_{+, \times}(\mathbf{I}) = |\mathbf{I}_{+, \diamond}^c - \mathbf{I}_{\times, \square}^c| \quad (12)$$

Results obtained using this last operator are shown in Figure 2(c). This is the operator that we propose to use as a corner detector. To extract corners from the output of $c_{+, \times}(\mathbf{I})$, it appeared to us that a simple global thresholding is sufficient even if this leads to multiple-pixel response for some corners (i.e. occurrence of a corner represented by a few connected pixels in the binary corner map). In fact, we found that non-maxima suppression, which is required in the other methods, does not improve the quality of the detection in the case of asymmetrical closing. In particular, non-maxima suppression does not rule out the multiple-pixel response mainly because the corner strength at these location are nearly equal (to the brightness difference between the corner and its background). However, this multiple pixel response behavior is not problematic in most applications and the elimination of the non-maxima suppression process reduces the computational load of the corner detection task.

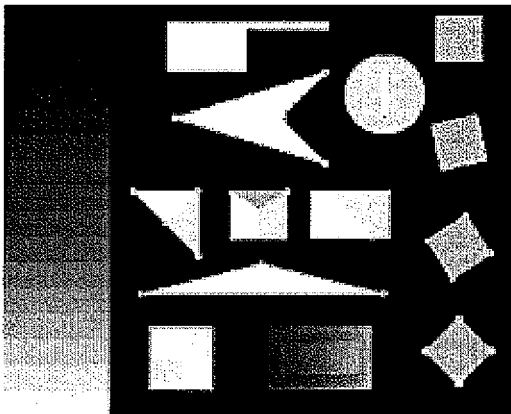
5 Comparative results

In order to test the validity of the operator $c_{+, \times}(\mathbf{I})$ as a corner detector, a series of tests were performed. Comparisons were made with the Plessey corner detector and the SUSAN corner detector.

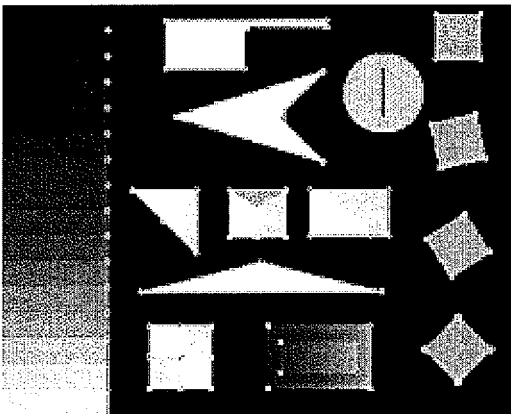
Sensitivity to noise was tested by adding a gaussian noise to the test image. The comparative results are shown in Figure 3. These tests demonstrate the superiority of methods based on brightness comparison



(a)



(b)



(c)

Figure 2: Corner detection by asymmetrical closing. (a) corner detected by c_+ . (b) corner detected by c_x . (c) corner detected by $c_{+,x}$.

over differential detectors. This observation has also been made in [17]. The SUSAN operator has also been reported to be 10 times faster than the Plessey operator. We observed that the method proposed here is about 2 times more efficient than the SUSAN operator.

To evaluate the stability of these detectors, we used a test sequence of 4 images showing a table from different points of view. The checkerboard pattern on the table creates 24 corners; the goal here is to test the ability of each operator to detect these 24 corners. Among these corners there are 4 L-junctions, 12 T-junctions and 8 X-junctions. All parameters in each method have been set in order to obtain the best possible results and a comparable density of corner points. For each method, the same parameter values are used for all images of the sequence. For the SUSAN operator and our operator based on asymmetrical closings, we have tested two different threshold values. Results are shown in Figure 4 where we show 2 of the 4 images. Table 1 presents the number of corners that each method has detected. It is not surprising to find, from the analysis of these results, that a lower threshold leads to a higher number of detected corners. But, at the same time, the number of false positives grows rapidly. In order to estimate the number of false positives that each method produced, we have counted the number of detected corners lying on the table in each image. The result of this analysis is presented in Table 2. These results demonstrate the reliability of the operator based on asymmetrical closings. It appears to be more stable while producing fewer false positives. It is also interesting to note that, in the case of the operator proposed in this paper, several of the false positive detections are due to the fact that X-junctions tend to produce a two-corner response.

6 Conclusion

We have presented a new operator for corner detection. It uses a variant of the morphological closing operator, which we have called asymmetrical closing. We found that this kind of approach, based on brightness comparisons, leads to better quality results than other approaches and is achieved at a lower computational cost. Stable and accurate corner detection has been obtained using the operator presented in this paper. Because of its algorithmic simplicity, we believe that this operator is an efficient means of producing input points of interest for feature-based approaches to 3D structure and motion estimation problems.

References

- [1] S.T. Barnard, W.B. Thompson, "Disparity analysis of images", *IEEE Trans. on PAMI*, 2:333-340,

1980.

- [2] J. Weng, N. Ahuja, T.S. Huang, "Matching Two Perspective Views", *IEEE Trans. on PAMI*, 14:806-825, 1992.
- [3] Z. Zhang, R. Deriche, O.D. Faugeras, Q.-T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar", *Artificial Intelligence Journal*, 78:87-119, 1994.
- [4] N.A. Thacker, J.E.W. Mayhew, "Optimal combination of stereo camera calibration from stereo images", *Image and vision computing*, 9:27-32, 1991
- [5] H. Assada, M. Brady, "The curvature primal sketch", *IEEE PAMI*, 8(1):2-14, 1986.
- [6] G. Medioni, Y. Yasumoto, "Corner detection and curve representation using cubic B-splines", *CVGIP*, 39:267-278, 1987.
- [7] R. Deriche, O.D. Faugeras, "2-D curves matching using high curvature points", *Proc. of the 10th IAPR*, pp. 18-23, 1990.
- [8] P.R. Beaudet, "Rotational invariant image operators", *Proc. Int. Conf on Pattern Recognition*, pp. 579-583, 1978.
- [9] L. Kitchen, A. Rosenfeld, "Gray-level corner detection", *Pattern Recognition Letters*, pp. 95-102, Dec. 1982.
- [10] L. Dreschler, H.H. Nagel, "Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scene", *CVGIP*, 20:199-228, 1982.
- [11] O.A. Zuniga, R.M. Haralick, "Corner detection using the facet model", *Proc. Int. Conf. on Pattern Recognition*, pp. 30-37, 1983.
- [12] R. Deriche, G. Giraudon, "Accurate corner detection: an analytical study", *Proc. Int. Conf. on Computer Vision*, pp.66-70, 1990.
- [13] H.P. Moravec, "Towards automatic visual obstacle avoidance", *Proc. Int. Joint Conf. on AI*, pp. 584-586, 1977.
- [14] C.G. Harris, "Determination of ego-motion from matched points", *Proc. Alvey Vision Conf.*, 1987.
- [15] J.A. Noble, "Finding corners", *Image and Vision Computing*, 6:121-128, 1988.

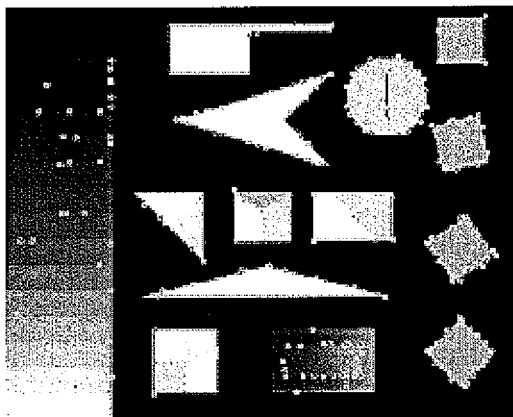
Plessey	87
SUSAN (threshold = 12)	88
SUSAN (threshold = 11)	90
asymmetrical closings (threshold = 10)	89
asymmetrical closings (threshold = 8)	95

Table 1: Number of corners correctly detected on the table. A total of 96 corners were considered in the test sequence, 24 in each image.

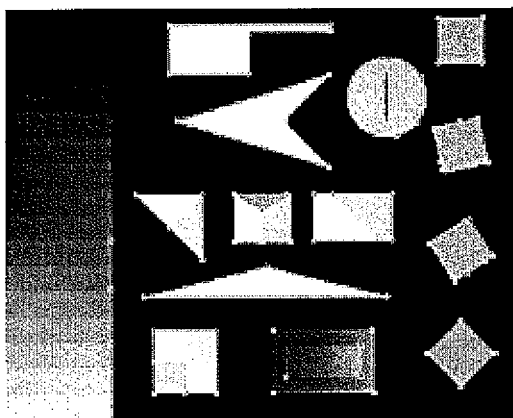
Plessey	221
SUSAN (threshold = 12)	163
SUSAN (threshold = 11)	183
asymmetrical closings (threshold = 10)	119
asymmetrical closings (threshold = 8)	157

Table 2: Number of detected corners on the table. This number should be equal to 96 in the ideal case.

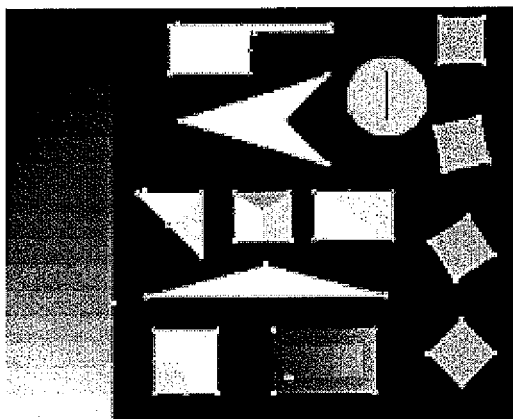
- [16] C. Harris, M. Stephens, "A combined corner and edge detector", *Proc. of 4th Alvey Vision Conference*, pp. 147-151, 1988.
- [17] S.M. Smith, J.M. Brady, "SUSAN - a new approach to low level image processing", *Int. Journal of Computer Vision*, 1997.
- [18] R. Deriche, T. Blaszk, "Recovering and characterizing image features using an efficient model based approach", *Proc. of CVPR*, pp. 530-535, 1993.
- [19] K. Rohr, "Modeling and identification of characteristic intensity variations", *Image and Vision Computing*, 10(2):66-76, 1992.



(a)

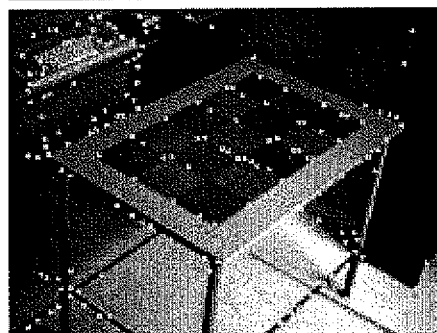
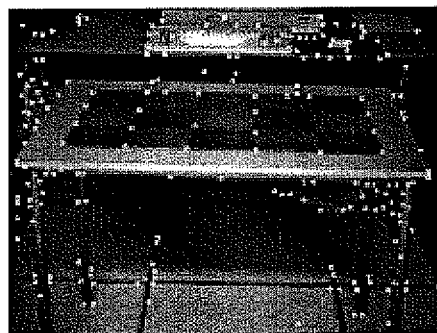


(b)

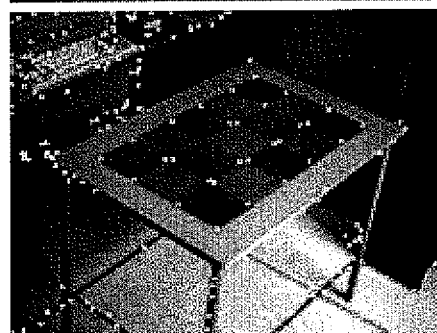
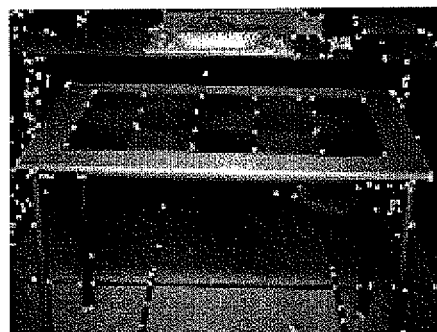


(c)

Figure 3: Test image with gaussian noise of variance $\sigma^2 = 50$. (a) Plessey. (b) SUSAN. (c) asymmetrical closing.



(a)



(b)

Figure 4: Corner detection on the test sequence. (a) The SUSAN operator (threshold = 12). (b) Asymmetrical closing (threshold = 10).



US006519350B1

(12) **United States Patent**
Van Overveld et al.

(10) **Patent No.:** **US 6,519,350 B1**
(45) Date of Patent: **Feb. 11, 2003**

(54) **EMBEDDING WATERMARKS IN IMAGES**

(75) **Inventors:** **Cornelis W. A. M. Van Overveld**,
 Eindhoven (NL); **Peter M. J. Rongen**,
 Eindhoven (NL); **Maurice J. J. B. Maes**,
 Eindhoven (NL)

(73) **Assignee:** **Koninklijke Philips Electronics N.V.**,
 Eindhoven (NL)

(*) **Notice:** Subject to any disclaimer, the term of this
 patent is extended or adjusted under 35
 U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** **09/477,220**

(22) **Filed:** **Jan. 4, 2000**

(30) **Foreign Application Priority Data**

Jan. 15, 1999 (EP) 99200123

(51) **Int. Cl.⁷** **G06K 9/66; G06K 9/00**

(52) **U.S. Cl.** **382/100; 382/193; 382/194;**
347/131; 347/136; 358/298; 358/296

(58) **Field of Search** **382/100, 193,**
382/194; 374/131, 136; 358/298, 296

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,530,759 A * 6/1996 Braudaway et al. 380/54
 5,825,892 A * 10/1998 Braudaway et al. 380/51
 5,831,657 A * 11/1998 Sakaue et al. 347/131

OTHER PUBLICATIONS

M.J.J.B. Maes and C.W.A.M. Van Overveld: "Digital Water-
 marking by Geometric Warping", Proceedings of the 1998
 International Conference on Image Processing, Oct. 4-7,
 1998, pp. 424-426.

C. Harris and M. Stephens, "A Combined Corner and Edge
 Detector", Proceedings of the 4th Alvery Vision Confer-
 ence, 1988, pp. 147-151.

* cited by examiner

Primary Examiner—Leo Boudreau

Assistant Examiner—Tom Y Lu

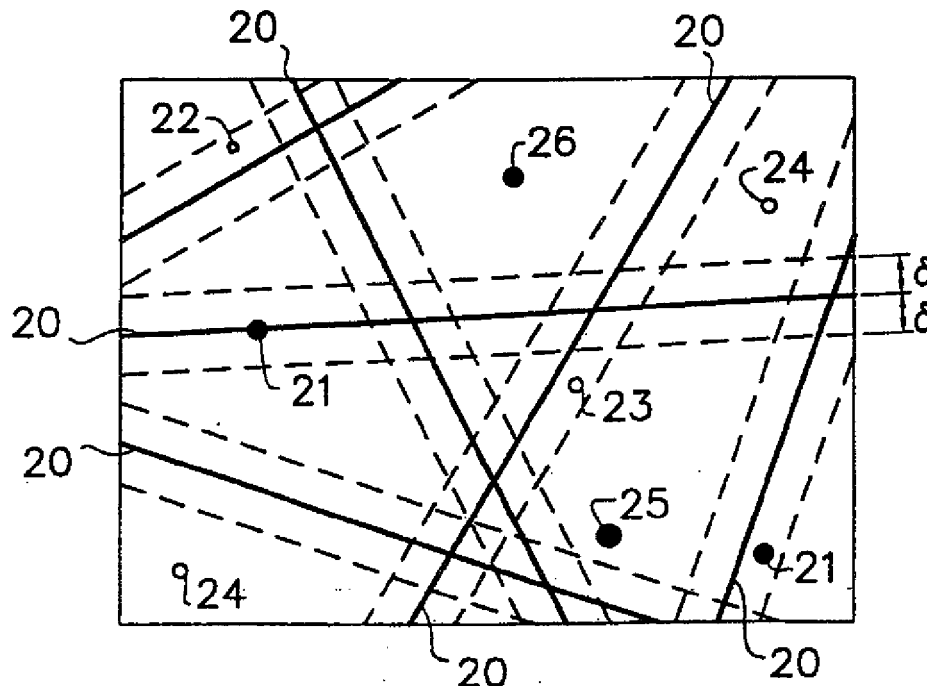
(74) *Attorney, Agent, or Firm*—Russell Gross

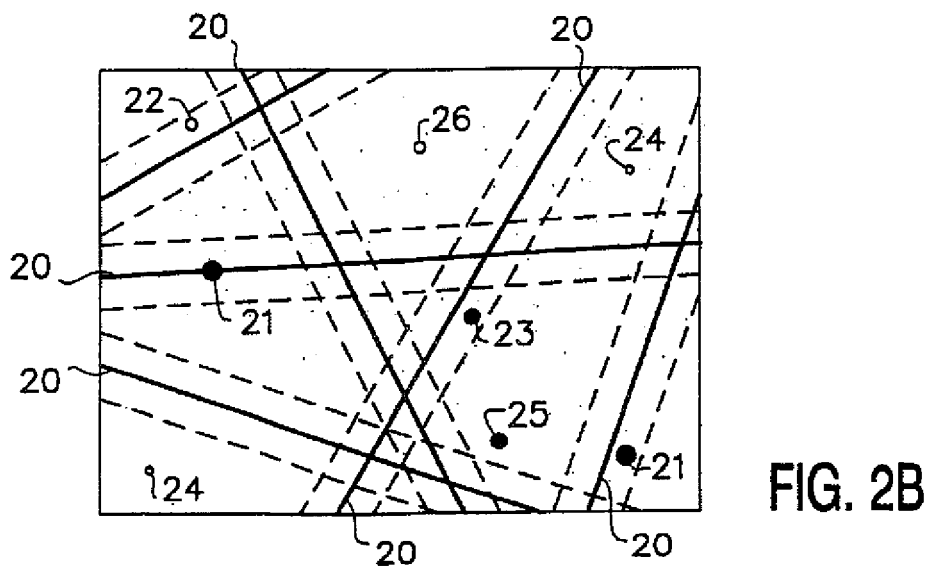
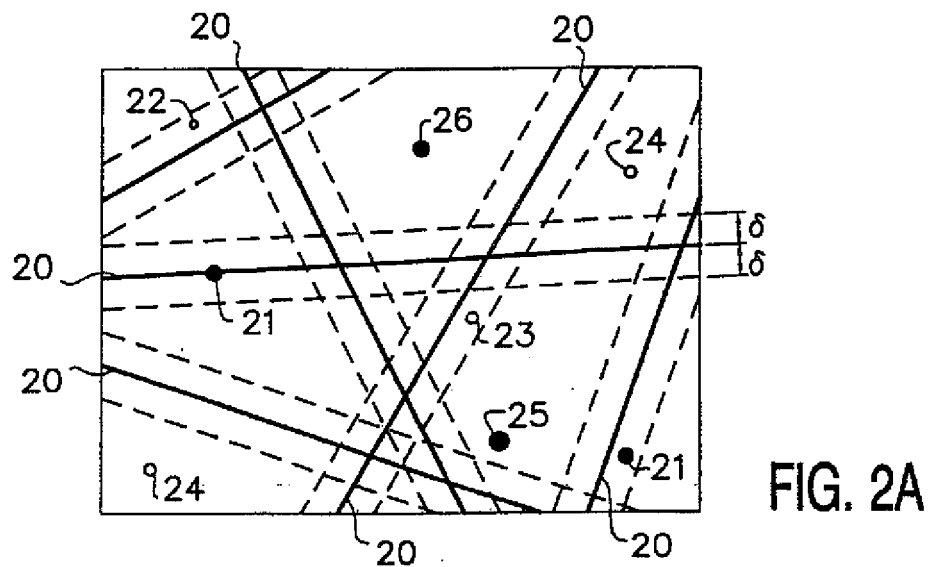
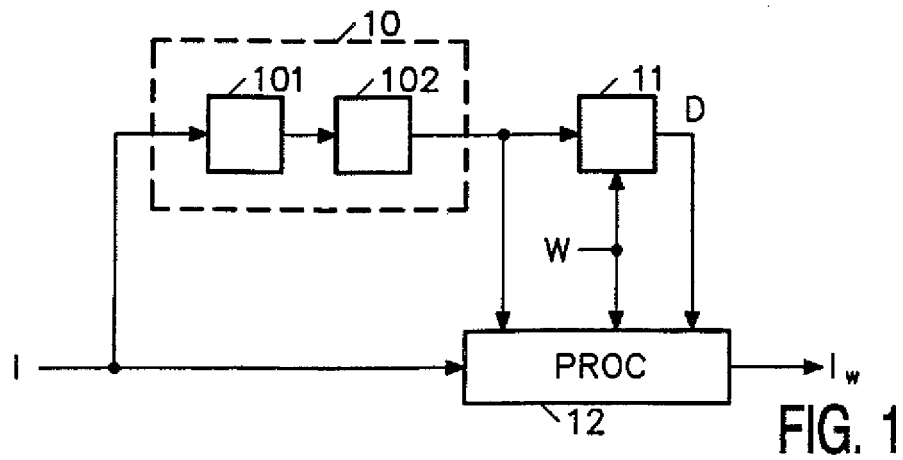
(57)

ABSTRACT

A method and arrangement for embedding a watermark in an
 image are disclosed. The watermark consists of a pseudo-
 random, dense subset of image pixels, e.g. a pattern of lines
 (20). A number of salient image pixels (21-26), for example,
 local extremes, corners or edges, is identified and it is
 determined whether they lie on (i.e. within a vicinity δ of)
 the line pattern (21-23) or not (24-26). In an unwatermarked
 image (FIG. 2A), the number of most salient pixels (21)
 lying on the watermark is substantially the same as the
 number of most salient pixels (25,26) not lying on the
 watermark. The image is watermarked (FIG. 2B) by modi-
 fying the saliency of the salient pixels in such a way that a
 significant majority (21,23) of the most salient pixels (21,
 23,25) is eventually located within the vicinity of the line
 pattern.

7 Claims, 3 Drawing Sheets





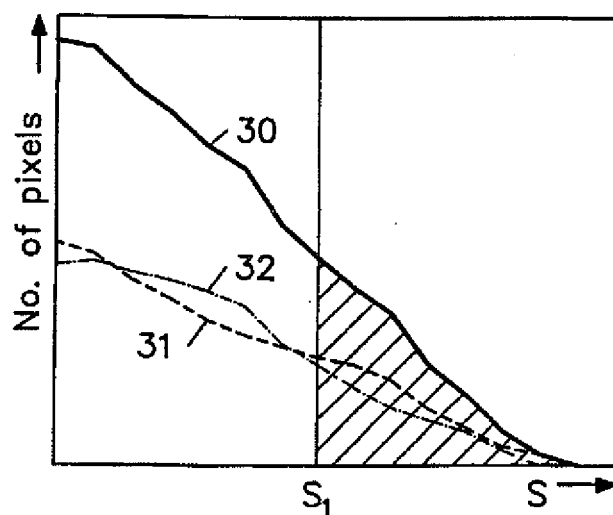


FIG. 3A

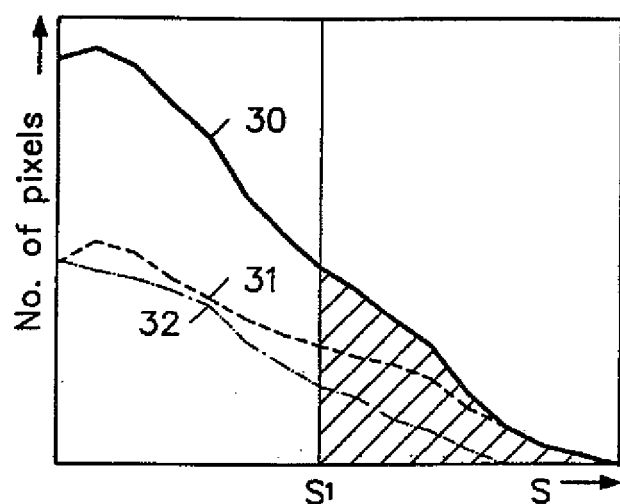


FIG. 3B

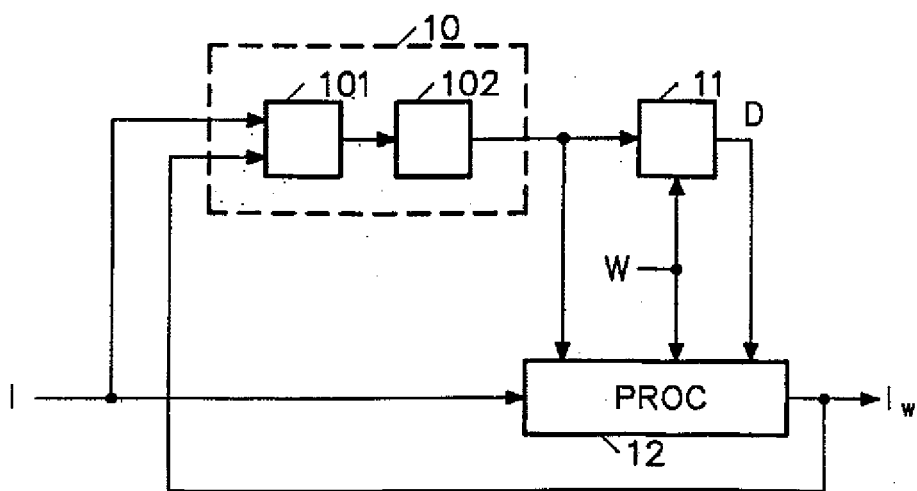


FIG. 4

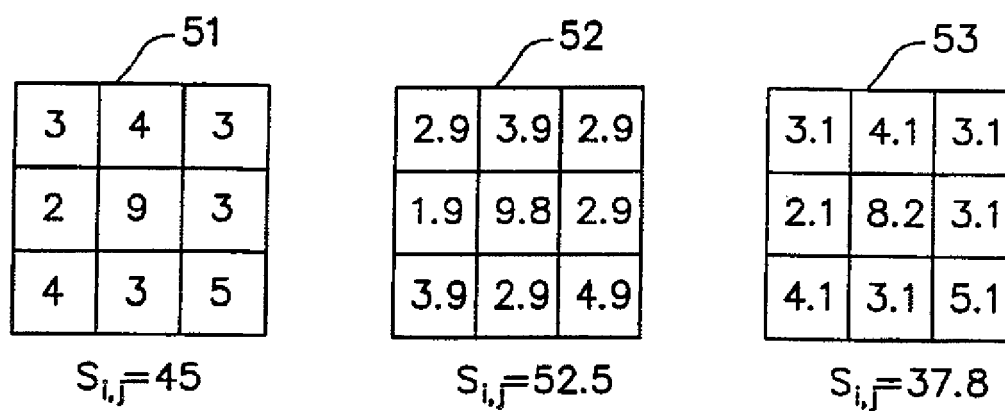


FIG. 5

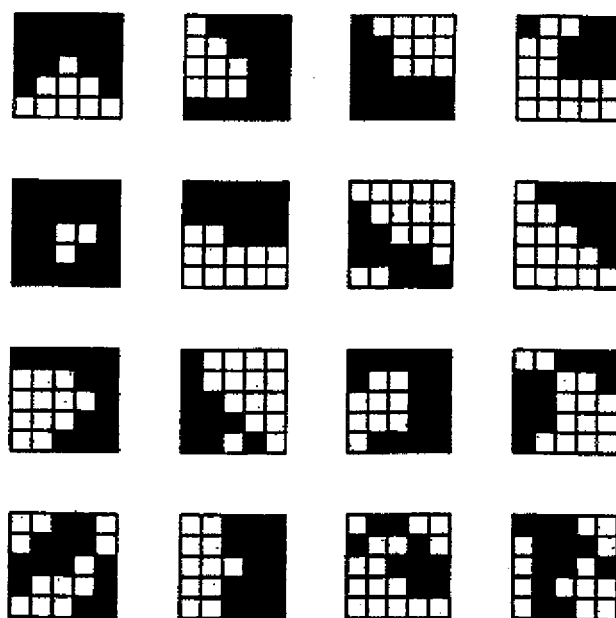


FIG. 6

EMBEDDING WATERMARKS IN IMAGES

FIELD OF THE INVENTION

The invention relates to a method of embedding a watermark in an image, comprising the steps of calculating a saliency of image pixels, identifying salient image pixels, and processing the image in such a way that a predetermined percentage of the most salient image pixels lies within the vicinity of a predetermined watermark pattern. The invention also relates to an arrangement for embedding a watermark in an image.

BACKGROUND OF THE INVENTION

Known method of embedding a watermark as defined in the opening paragraph is disclosed in M. J. J. B. Maes and C. W. A. M. van Overveld: "Digital Watermarking by Geometric Warping", Proceedings of the 1998 International Conference on Image Processing, Oct. 4-7, 1998, pages 424-426. In this known method, the watermark is a predetermined image pattern, for example, a pattern of lines. The image is watermarked if a statistically high percentage of salient pixels of the image lies within the vicinity of the watermark pattern. This is achieved by identifying the salient pixels, and moving ("warping") them to the vicinity of the watermark pattern. The step of image processing thus comprises locally changing the geometrical characteristics of the image.

OBJECT AND SUMMARY OF THE INVENTION

It is an object of the invention to embed the watermark using an alternative method of image processing.

To this end, the method in accordance with the invention is characterized in that the step of processing the image comprises modifying the saliency of salient pixels. The watermark is thus embedded by modifying the saliency of image pixels instead of moving them to different positions. The saliency is modified by decreasing the saliency of most salient pixels not lying within the vicinity of the watermark pattern and/or increasing the saliency of salient pixels lying within the vicinity of the watermark pattern. Advantageous embodiments of identifying and modifying salient pixels are defined in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows schematically an arrangement for embedding a watermark in an image in accordance with the invention.

FIGS. 2A and 2B show watermark patterns and salient pixels to illustrate the operation of the arrangement which is shown in FIG. 1.

FIGS. 3A and 3B show histograms to illustrate the operation of the arrangement which is shown in FIG. 1.

FIG. 4 shows a further embodiment of the arrangement for embedding a watermark in an image in accordance with the invention.

FIGS. 5 and 6 show sub-images to illustrate the operation of alternative options for identifying and modifying salient pixels.

DESCRIPTION OF EMBODIMENTS

FIG. 1 shows an embodiment of an arrangement for embedding a watermark in an image in accordance with the invention. The arrangement comprises a salient point extrac-

tion module 10, a decision module 11, and an image-processing module 12. The arrangement receives an input image I and a watermark W, and generates a watermarked image I_w .

FIG. 2A shows an example of the watermark. In this example, the watermark W is assumed to be a pattern of lines 20, but this is not essential. Salient pixels are shown as circles 21-26 in the Figure. The diameter of a circle represents the saliency of the pixel. A pixel is said to lie in the vicinity of the watermark, if the distance from that pixel to the nearest line 20 is less than a predetermined value. This predetermined distance may be a fixed value δ . This definition of vicinity, where the lines 20 have a "thickness" 2δ , will be adhered to throughout the rest of the description. The expression "lie in the vicinity of the watermark" will often also be referred to as "lie on the watermark". In FIG. 2A, the pixels 21-23 lie on the watermark, whereas pixels 24-26 do not.

Alternatively, the expression "lie in the vicinity of the watermark" can be defined to mean that the saliency-to-distance ratio has a predetermined minimum value. Thus, pixel 25 in FIG. 2A may lie in the vicinity of the watermark whereas, say, pixel 22 does not, because the larger distance of pixel 25 to the nearest line 20 is compensated by its larger saliency.

The watermark pattern W has a given density. This is understood to mean that the watermark pattern including its vicinity covers a given portion (p) of the image. Hereinafter, it will be assumed that the watermark covers $p=50\%$ of the image. Accordingly, about $p=50\%$ of the salient pixels of an unwatermarked image will lie on the watermark.

The arrangement shown in FIG. 1 operates as follows. The salient pixel extraction module 10 comprises saliency-calculating means 101 for calculating the saliency of the image pixels. Embodiments thereof will be described later. The module 10 further comprises a selection circuit 102, which selects salient pixels. The number of salient pixels is small compared with the total number of image pixels. The salient pixels are reasonably uniformly distributed across the image, so as to avoid that modifying the saliency of a pixel affects an earlier change of a nearby salient pixel. This can be achieved, inter alia, by dividing the image into blocks and selecting one or a few salient pixels per block. It can also be achieved by requiring a minimal distance between salient points. As already mentioned, the pixels 21-26 in FIG. 2A are salient pixels.

The salient pixels and their saliencies are applied to the decision module 11. This module receives the watermark pattern W and determines which portion of the most salient pixels lie on the watermark. In an unwatermarked image, a percentage p (here $p=50\%$) of the most salient pixels will lie on the watermark. In FIG. 2A, the most salient pixels are shown as solid circles. Of these most salient pixels, the pixels 21 lie on the watermark whereas the pixels 25 and 26 do not. If the percentage of the most salient pixels lying on the watermark W does not substantially differ from $p=50\%$, the decision module 11 generates a decision signal $D=0$. In response to this signal, the image-processing module 12 modifies the saliency of the salient pixels in such a way that a significant percentage of the most salient pixels will lie on the watermark. Note that the extraction module 10 and decision module 11 jointly constitute a watermark detector.

The image-processing module 12 decreases the saliency of salient pixels not lying on the watermark and/or increases the saliency of salient pixels lying on the watermark. FIG. 2B shows the saliencies of the salient pixels 21-26 after

3

modification. The saliencies of the pixels 21–23 lying on the watermark are increased, the saliencies of the pixels 24–26 not lying on the watermark are decreased. The process of modifying turns “nearly most salient pixels” into most salient pixels and vice versa. In FIG. 2B, the pixel 23 lying on the watermark is now one of the most salient pixels and thus shown as a solid circle. The pixel 26 not lying on the watermark is no longer one of the most salient pixels. Thus, three of the four most salient pixels lie on the watermark after the process of modification.

FIGS. 3A and 3B illustrate the process in the form of histograms. Reference numeral 30 is a graph of the number of salient pixels versus saliency. Numeral 31 denotes the pixels lying on the watermark pattern W, numeral 32 denotes pixels not lying on the watermark. The shaded area denotes the set of most salient pixels. In this example, the set includes all pixels having a saliency which is larger than a given value S_1 . Alternatively, the set may have a predetermined number of most salient pixels. FIG. 3A shows the histogram of an unwatermarked image, having 55% of the most salient pixels lying on the watermark. FIG. 3B shows the histogram after increasing the saliency of pixels lying on the watermark (which shifts line 31 to the right) and decreasing the saliency of pixels not lying on the watermark (which shifts line 32 to the left). After this step, 70% of the most salient pixels lies on the watermark.

In a preferred embodiment of the arrangement, the process of modifying saliency may be repeated until a predetermined majority, e.g. 75%, of the most salient pixels has been found to lie on the watermark. Such an embodiment is shown in FIG. 4. The arrangement differs from the one which is shown in FIG. 1 in that the processed image I_w is fed back to the extraction module 10 until a significant percentage of the most salient pixels has been found to lie on the watermark, and the decision module 11 generates the signal D=1.

The step of increasing the saliency of a pixel, which is carried out by the image-processing module 12, implies locally adding a luminance and/or chrominance image ΔI to the image I in such a way that the saliency S is amplified. Similarly, decreasing the saliency of a pixel implies adding a luminance and/or chrominance image ΔI to the image I in such a way that the saliency S is attenuated. In view thereof, it will be appreciated that the method of modifying is strongly related to the method of calculating the saliency.

In one embodiment of the arrangement in accordance with the invention, a 2-dimensional filter forms the saliency-calculating means 101. Such a filter will hereinafter be represented by a matrix F , for example, the following 3*3 matrix:

$$F = \begin{bmatrix} f_{-1,-1} & f_{-1,0} & f_{-1,1} \\ f_{0,-1} & f_{0,0} & f_{0,1} \\ f_{1,-1} & f_{1,0} & f_{1,1} \end{bmatrix}$$

The saliency $S_{i,j}$ of pixel $I_{i,j}$ (where i and j denote the vertical and horizontal pixel positions, respectively) is defined by the following equation:

$$S_{i,j} = f_{-1,-1}I_{i-1,j-1} + f_{-1,0}I_{i-1,j} + f_{-1,1}I_{i-1,j+1} + f_{0,-1}I_{i,j-1} + f_{0,0}I_{i,j} + f_{0,1}I_{i,j+1} + f_{1,-1}I_{i+1,j-1} + f_{1,0}I_{i+1,j} + f_{1,1}I_{i+1,j+1} \quad (\text{Eq. 1})$$

In one embodiment of the invention, the filter F is a Laplace filter. This is a high-pass filter which returns values that are indicative of local minima and maxima of the pixel values. It returns the value zero if the pixel value $I_{i,j}$ is equal to the

4

average of its neighbors. The 3*3 Laplace filter is represented by the matrix:

$$F = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

In this embodiment, the saliency $S_{i,j}$ of a pixel is modified by adding a weighted version of the matrix F to the image. In mathematical notation:

$$I_m = I + \lambda F \text{ to increase the saliency,} \quad (\text{Eq. 2})$$

and

$$I_m = I - \lambda F \text{ to decrease the saliency.} \quad (\text{Eq. 3})$$

where λ is a given weighting factor, I is a 3*3 sub-image having the salient pixel in the center, and I_m is the modified 3*3 sub-image.

FIG. 5 shows an example of this modification process. Numeral 51 denotes a 3*3 sub-image with a salient pixel having an intensity $I_{i,j}=9$ in the center. The pixel has a saliency $S_{i,j}=45$ in accordance with Eq. 1. Numeral 52 denotes the sub-image after processing in accordance with Eq. 2 and $\lambda=0.1$, which increases the saliency to $S_{i,j}=52.5$. Numeral 53 denotes the sub-image after processing in accordance with Eq. 3 and $\lambda=0.1$, which decreases the saliency to $S_{i,j}=37.8$.

An alternative method of increasing the saliency $S_{i,j}$ is based on the recognition that $S_{i,j}$ is already large, and that the sub-image I itself may be used to amplify the saliency, i.e.: $I_m = I + \lambda I$.

In another embodiment of the arrangement, the saliency-calculating means 101 is formed by an edge and/or corner detector, the saliency of a pixel being represented by the edge or corner strength. Corner detectors are known per se. An advantageous embodiment is described in C. Harris and M. Stephens: “A Combined Corner and Edge Detector”, Proceedings of the 4th Alvey Vision Conference, 1988, pages 147–151. This corner detector is defined by a matrix:

$$M = w(x, y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

where

$$w(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

is a Gaussian function with standard deviation σ , the symbol $*$ denotes convolution, and

$$I_x = \frac{\partial I}{\partial x} \quad \text{and} \quad I_y = \frac{\partial I}{\partial y}$$

are the partial derivatives of the image in the directions x and y , respectively. The matrix M can be relatively easily calculated by using the following discrete approximations:

$$w(x, y) = \frac{1}{13} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

-continued

$$I_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I, \text{ and}$$

$$I_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I.$$

The matrix M, which can be written in the form

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix},$$

has a determinant $D=AB-C^2$ and a trace $T=A+B$. The corner strength R is now defined by:

$$R=D-kT^2$$

where k is a suitable constant, for example, $k=0.01$. The corner strength R is positive for a corner, negative for an edge, and approximately zero in a flat region.

FIG. 6 shows some examples of 5*5 sub-images, the center pixels of which were found to have a large corner strength. Although the pixels have multi-bit luminance and chrominance levels, the sub-images are here shown as binary images, that is, pixels having intensities larger than a mean or median value are shown in white and pixels having intensities less than said mean or median value are shown in black. Note that for some sub-images it is immediately clear that the center pixel is indeed a corner, whereas for some it is not.

Amplifying the corner strength is achieved by increasing the contrast between the pixels that represent the corner and the complementary pixels, for example, by adding an amount ΔI to the intensities of the white pixels in FIG. 6 and/or subtracting an amount ΔI from the black pixels in FIG. 6. Weakening the corner strength is obtained by the inverse operation, i.e. subtracting ΔI from the white pixels and adding ΔI to the black pixels.

In the foregoing, it has been assumed that an image is watermarked if a significant percentage of salient pixels lies in the vicinity of the watermark pattern. It will be appreciated, however, that the complementary definition may be used in practice, i.e. that an image is watermarked if a significant percentage of salient pixels lies outside the vicinity of the watermark pattern.

It is further noted that insufficient salient pixels may be found in certain areas of the input image. This may be particularly the case in uniform areas of synthetic images such as cartoons. In this case, salient pixels lying on the watermark can be created, inter alia, by adding particular noise patterns to said areas of the image, by adding intensities corresponding to the filter coefficients of the Laplace filter, or by artificially creating corners.

In summary, a method and arrangement for embedding a watermark in an image are disclosed. The watermark consists of a pseudo-random, dense subset of image pixels, e.g. a pattern of lines (20). A number of salient image pixels (21-26), for example, local extremes, corners or edges, is identified and it is determined whether they lie on (i.e. within a vicinity δ of) the line pattern (21-23) or not (24-26). In an unwatermarked image (FIG. 2A), the number of most salient pixels (21) lying on the watermark is substantially the same as the number of most salient pixels (25,26) not lying on the

watermark. The image is watermarked (FIG. 2B) by modifying the saliency of the salient pixels in such a way that a significant majority (21,23) of the most salient pixels (21, 23,25) is eventually located within the vicinity of the line pattern.

What is claimed is:

1. A method of embedding a watermark in an image, comprising the steps of calculating (101) a saliency (S_{ij}) of image pixels, identifying (102) salient image pixels (21-26), and processing the image in such a way that a predetermined percentage of the most salient image pixels (21,25,26) lies within the vicinity (δ) of a predetermined watermark pattern (20), wherein said step of image processing includes modifying (12) the saliency of salient pixels (23,26) by decreasing the saliency of most salient pixels not lying within the vicinity of the watermark pattern.

2. A method of embedding a watermark in an image, comprising the steps of calculating (101) a saliency (S_{ij}) of image pixels, identifying (102) salient image pixels (21-26), and processing the image in such a way that a predetermined percentage of the most salient image pixels (21,25,26) lies within the vicinity (δ) of a predetermined watermark pattern (20), wherein said step of image processing includes modifying (12) the saliency of salient pixels (23,26), wherein the step of calculating the saliency of pixels includes filtering the image, and the step of image processing includes changing the pixel intensities of a sub-image including a salient pixel to modify the response of said filter in accordance with a desired modification of the saliency of said salient pixel.

3. A method as claimed in claim 2, wherein the filter is a 2-dimensional filter, and the step of modifying the saliency comprises creating a linear combination of the sub-image and the filter coefficients of said filter.

4. A method as claimed in claim 2, wherein the filter is a corner detection filter the response of which represents a corner strength, and the step of modifying comprises changing the pixel intensities of the sub-image to modify said corner strength.

5. A method as claimed in claim 1, further comprising the step of feeding back the processed image and repeatedly carrying out the steps of calculating, identifying and processing until the significant percentage of the most salient image pixels lies within the vicinity of the watermark pattern.

6. A method of embedding a watermark in an image, comprising the steps of calculating (101) a saliency (S_{ij}) of image pixels, identifying (102) salient image pixels (21-26), and processing the image in such a way that a predetermined percentage of the most salient image pixels (21,25,26) lies within the vicinity (δ) of a predetermined watermark pattern (20), creating salient pixels in uniform image areas by adding predetermined pixel patterns to said areas, wherein said step of image processing includes modifying (12) the saliency of salient pixels (23,26).

7. An arrangement for embedding a watermark (W) in an image (I), comprising means (101) for calculating a saliency (S_{ij}) of image pixels, means (102) for identifying salient image pixels, and means (12) for processing the image in such a way that a predetermined percentage of the most salient image pixels lies within the vicinity (δ) of a predetermined watermark pattern (20), wherein said image-processing means (12) is arranged to modify the saliency of salient pixels by decreasing the saliency of most salient pixels not lying within the vicinity of the watermark pattern.

* * * * *